DOCTEUR EN SCIENCES

A parallelized micro-simulation platform for population and mobility behavior. Application to Belgium.

Barthelemy, Johan

Award date:
2014

Awarding institution:
Universite de Namur

UNIVERSITÉ DE NAMUR

FACULTÉ DES SCIENCES

DÉPARTEMENT DE MATHÉMATIQUE

A parallelized micro-simulation platform for population and mobility behaviour

Application to Belgium

Thèse présentée par
Johan Barthélemy
pour l'obtention du grade
de Docteur en Sciences

Composition du Jury:

Philippe Toint (Promoteur)
Eric Cornelis (Co-promoteur)
Timoteo Carletti
Guillaume Deffuant
Bart Jourquin
Anne Lemaitre (Présidente du Jury)

Avril 2014

Imprimé en Belgique

ISBN : 978-2-87037-842-7
Dépôt légal: D / 2014 / 1881/ 32

Université de Namur
Faculté des Sciences
rue de Bruxelles, 61, B-5000 Namur (Belgique)

Une plateforme de micro-simulation de populations et du comportement de mobilité adaptée au calcul parallèle
Application à la Belgique
par Johan Barthélemy

Résumé : L'objectif de cette thèse est la conception d'une plateforme de micro-simulation de populations et du comportement de mobilité. Le micro-simulateur repose sur la méthodologie dite par agent et exploite les techniques de calculs parrallèle afin de traiter des populations de grande taille. En particulier, la génération d'agents et la modélisation de leur mobilité pour la Belgique seront abordés. La première étape consiste à développer un générateur de populations synthétiques dont les caractéristiques principales sont la non-utilisation d'un échantillon significatif de la population et sa capacité à gérer des incohérences dans les données disponibles. La demande de transport est alors générée par un modèle stochastique basé sur les chaînes d'activités et qui ne requiert qu'une quantité limitée de données. Finalement cette demande est affectée sur le réseau routier au moyen d'une approche comportementale reposant sur le concept d'agents stratégiques, et non pas via un modèle classique d'affection dynamique du trafic basée sur des méthodes de simulations.

A parallelized micro-simulation platform
for population and mobility behaviour
Application to Belgium
by Johan Barthélemy

Abstract: This thesis aims at developing an agent-based micro-simulation framework for (large) population evolution and mobility behaviour. More specifically we focus on the agents generation and the traffic simulation parts of the platform, and its application to Belgium. Hence we firstly develop a synthetic population generator whose main characteristics are its sample-free nature, its ability to cope with moderate data inconsistencies and different levels of aggregation. We then generate the traffic demand forecasting with a stochastic and flexible activity-based model relying on weak data requirements. Finally, a traffic simulation is completed by considering the assignment of the generated demand on the road network. We give the initial developments of a strategic agent-based alternative to the conventional simulation-based dynamic traffic assignment models.

Thèse de doctorat en Sciences Mathématiques (Ph.D. thesis in Mathematics)
Date: 25/04/2014
Département de Mathématique
Promoteurs (Advisors): Prof. Ph. Toint et Dr. E. Cornelis

Remerciements

Enfin ! Une étape de près de six ans et demi qui se termine ! Un achèvement qui n'aurait jamais été possible sans les nombreuses personnes qui ont été présentes à mes côtés pour partager les meilleurs moments mais aussi m'épauler lors des situations plus difficiles.

J'aimerai tout d'abord remercier mes deux promoteurs, Eric Cornélis et Philippe Toint, non seulement pour m'avoir offert la chance de faire une thèse, mais surtout pour leur encadrement, la confiance qu'ils m'ont accordée et la liberté qu'ils m'ont laissée tout au long de ce travail. Les discussions que nous avons eues ensemble ainsi que leurs conseils et leurs avis éclairés m'ont toujours permis de trouver une solution à chacun des problèmes rencontrés. Grâce à eux, j'ai pu côtoyer de nombreuses personnes intéressantes à travers le monde, découvrir des sujets passionnants et donner le meilleur de moi-même pour obtenir le résultat que vous tenez entre vos mains !

Merci tout particulièrement à Timoteo Carletti pour sa collaboration à une partie de ce travail et pour avoir accepté d'être dans le jury de cette thèse. Mes remerciements vont également aux autres membres de ce jury : Anne Lemaître, Guillaume Deffuant et Bart Jourquin. Leurs remarques pertinentes et leurs suggestions ont permis d'approfondir et d'améliorer de nombreux aspects de cet ouvrage.

Ma famille a incontestablement joué un rôle important durant toutes ces années, par leur présence, leur soutien indéfectible et l'intérêt qu'ils ont toujours porté à mon travail. Merci à Maman pour toutes ses attentions (notamment culinaires), à Papa pour avoir toujours été là et m'avoir permis de *tripatouiller* sa moto et ses ordinateurs pour me changer les idées, à Odile pour sa gentillesse et son écoute à toutes heures du jour et de la nuit, à Bastien pour les soirées tranquilles à l'appartement et son calme en toutes circonstances, Nicolas pour tous les barbecues et les sessions bricolages, Mémé pour toutes les tartes, sa bonne humeur et ses coups de téléphones à l'improviste, à Bobonne et Bon Papa, partis trop tôt mais dont le souvenir ne me quitte jamais. Merci aussi à tante Nadine, Dominique, Jean-Marie, Yannick, Grégory, Martine, Catherine, Sarah, Raphaël, Gana, Stéphane et Dgilnoye. Merci du fond du cœur d'avoir toujours cru en moi !

Evidemment, mes plus chaleureux remerciements s'adressent à tous mes amis, sur lesquels j'ai toujours pu compter ! Grâce à vous ces dernières an-

nées sont passées très (trop ?) vite. Merci René pour ces presque trente années d'amitié ! Je tiens à remercier Sandra, Delphine, Frip, Rosa et Alice pour leur accueil aux facs il y a dix ans et demi, Céline pour tous les bons moments passé ensemble pendant, et après, nos études, et Audrey, celle qui s'est occupée de moi comme une petite maman lors de la fin de la thèse. Merci également à Sebastian pour les sessions de remises en forme indispensables ! Je remercie tout spécialement Patrick, Jehan et bien sûr Manu pour toutes les sorties faites ensemble, les soirées passées à refaire le monde (à Namur ou ailleurs), les discussions jusqu'au bout de la nuit et enfin tous nos délires mathématiques ! Quelle chance j'ai eu de tous vous rencontrer !

Pendant toutes ces années, j'ai eu le bonheur de partager le grand bureau du GRT, non pas simplement avec des collègues, mais de véritables amis. Pour cela, je remercie vivement Marie, Fabien (désolé pour le pari que je t'ai fait perdre), Xavier (qui a toujours su trouver les mots pour me motiver), Julien, Patrick, Laurie (ah le Brésil restera un moment fort de cette thèse), Marie et Véronique pour l'ambiance chaleureuse qu'il y a toujours eu de l'autre côté de la passerelle. Ça va être difficile de retrouver un bureau comme ça !

Je tiens également à exprimer toute ma reconnaissance à chacun des membres du Département de Mathématique. Ce fût un réel plaisir de partager votre quotidien et de travailler au sein de ce département ! En particulier, je remercie Marcel Rémon et André Hardy pour m'avoir donné le goût des statistiques quand j'étais étudiant et Annick Sartenaer pour ses encouragements lors de la dernière (longue) ligne droite. Merci à nos secrétaires efficaces que sont Pascale et Martine, ainsi qu'à Frédéric Wautelet qui a su répondre à mes nombreuses demandes concernant le cluster. Merci à Benoît pour ses bons mots et ses visites surprises au GRT. Le premier étage, et plus spécialement Jonathan (avec ses râleries légendaires), Jérémy (et sa geek attitude trollesque), Jérémy (et son coté anglais) et Martin (toujours prêt à partager ses goûts musicaux) ont toujours été là pour me faire rire... merci les jeunes ! Merci aux anciens doctorants du département, Julien, Nicolas, Sebastian, Charles, Patrick, Jehan, Audrey et Emilie, pour m'avoir montré qu'il était possible de venir au bout d'une thèse !

Pour terminer, j'aimerais remercier Pascal Perez et toute son équipe pour m'avoir accueilli en 2012 au sein de SMART dans le cadre d'un séjour de recherche, et qui m'a finalement proposé de les rejoindre deux années plus tard.

Bref... j'ai fait une thèse ! Merci à tous d'y avoir contribué, de près ou de loin, chacun à votre manière ! Maintenant, en route pour une nouvelle aventure australienne ! Place aux kangourous, aux koalas et aux coups de soleil !

Cheers !

Johan

Contents

Introduction

It is far better to foresee even without certainty than not to foresee at all.

– Henri Poincaré

Transportation plays a key role in societies. In developed and developing countries, a large majority of individuals is travelling every day to perform daily activities and exchange goods. As a result the transportation system of a country is closely related to development of its economy by meeting travel demand of people and allowing the transport and the exchange of resources (Mathew and Krishna Rao, 2007). Nevertheless transportation is also endowed with negatives effects: growth of transportation demand can lead to an increase of accidents, environmental issues such as air and noise pollutions and energy consumption.

Understanding the dynamics of transportation system has then naturally become a major research field. The computer assisted transport simulation has a long history since firsts attempts when developed in (Mathewson et al., 1955). Since a couple of decades these tools have become widely accepted amongst the research community and countless applications are grounded on such tools (Pursula, 1999) including transportation network optimization, land use planning, decision making for public transport policy, environmental quality improvement, transport demand forecasting and evaluating future infrastructure improvements.

This thesis is concerned with the development of a traffic simulation within the VirtualBelgium framework, an agent-based platform aiming at developing understanding of the evolution of the Belgian population using agent-based simulations and considering various aspects of this evolution such as demographics, residential choices, activity patterns, mobility, *etc.*

Agent-based systems

Complex systems characterized by many interacting actors occur everywhere in our world, and mobility behaviour of individuals is clearly one of them. The

complexity arises from the actors' interactions in their environment which often result in non-linear and emergent behaviours difficult to predict (Bazghandi, 2012). In other words, the whole system is more than the sum of its parts.

The recent advances in high performance and distributed computing paved the way to efficient models and simulations for these systems using agent-based models, also referred in this work as micro-simulation. The base unit of these models, the *agent*, represents an actor of the considered system. Even though there is no universally agreements on the term agent, Wooldridge (2008) proposed the following definition.

> An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objective.

These models have various advantages helping us to enhance our understanding of a considered system (van Dam et al., 2012):

- they provide a natural description of the system by focusing on its actor;
- they are able to capture emergent phenomena and behaviours;
- and experiments are conducted *in silico*, a more cost-effective and time saving approach than conventional experimentation.

Nevertheless there are challenging questions associated with agent-based models including agents generation, agents behaviour description and extreme computational requirements. These issues will be discussed in this thesis for building our VirtualBelgium simulation framework.

State of the art review

As this work focuses on slightly different topics (agents generation, traffic demand generation and assignment), we will present a review of the state of the art of these specific topics at beginning of appropriate chapters.

Structure of the thesis

This document is divided into four chapters. Firstly Chapter 1 introduces the VirtualBelgium's base architecture, agents characteristics and gives an overview of its structure.

Chapter 2 is then devoted to the agents creation with a sample-free synthetic population generator. The presentation of the generation algorithm and its application to the Belgian case is followed by convincing validation result.

In Chapter 3 we detail a flexible and stochastic activity-based model for generating travel demand and designed for nation-wide applications. The model is formally described before putting it into practice with the Belgian synthetic

population. Initial assessment of the model's performance is then illustrated by comparing the agents' travel demand to real world observations.

Traffic flow simulation is addressed in Chapter 4, which describes a dynamic traffic assignment model with strategic agents. The strategy, which gives to the agents the ability to adapt their path with respect to perceived local traffic conditions, is initially detailed and promising preliminary results are presented.

We finally conclude and give some perspectives of future works.

Contributions

Key results appearing in this document are the subject of three different papers: the first one have been published in a peer-reviewed journal, the others are submitted:

- J. Barthélemy and Ph. L. Toint. Synthetic population generation without a sample. *Transportation Science*, **47**(2), 266–279, 2013.
- J. Barthélemy and Ph. L. Toint. A stochastic and flexible activity-based model for large population. Application to Belgium. Submitted in *The Journal of Artificial Societies and Social Simulation* (February 2014).
- J. Barthélemy and T. Carletti. A dynamic traffic assignment model with strategic agents. *In preparation*

Core foundations of the VirtualBelgium framework and the models presented in this thesis are implemented in an open source project hosted on the SourgeForge platform (`http://sourceforge.net`). The compatibility of the produced code with existing open source tools has been a major concern. These implementation aspects constitute a significant part of the thesis contributions.

Chapter 1

VirtualBelgium: a micro-simulator for Belgium

Contents

1.1 Introduction

This first technical chapter aims at introducing the core elements of the VirtualBelgium agent-based micro-simulator in which the models developed in this thesis are exploited.

The next section is dedicated to the description of the agents involved in the VirtualBelgium micro-simulator and their environment. Then the organisation of the traffic and socio-demographic modules is presented in Section 1.3. Section 1.4 briefly introduces the platform and its foundations. We finally give in Section 1.5 an overview of the structural design of VirtualBelgium before detailing its compatibility with an agent-based framework for traffic simulation in Section 1.6.

1.2 Agents and environment description

VirtualBelgium is an agent-based micro-simulator, focusing on simulating the mobility behaviours and the demographic evolution of the Belgian population. Therefore the agents will be individuals gathered in households. These households will be located in one of the 589 municipalities of Belgium displayed in Figure 1.1. The individual agents will also evolve in an environment, namely the Belgian municipalities and road network.

The spatial data used to generate this Belgian environment is extracted from the OpenStreetMap project (Haklay and Weber, 2008). This collaborative and open-source project aims to create a free editable map of the world and provides contents under the Open Database License[(1)]. It provides free downloads of spatial data from a large selection of themes, including roads, transit lines, bicycle and pedestrian paths, tourist sites and land use layers. These data have been favourably compared with proprietary datasources (Zielstra and Hochmair, 2012).

As agent's attributes, we have chosen characteristics wich are known to significantly influence travel behaviour (Avery, 2011, Hubert and Toint, 2002, Cornelis et al., 2012). Individuals' and households' attributes are presented respectively in Tables 1.1 and 1.2. The generation process of these agents with these attributes is fully described in Chapter 2.

1.3 VirtualBelgium's modules

The mobility patterns of a given individual evolves together with his/her socio-demographic characteristics. Therefore it is interesting to implement processes representing an evolution process for the population of interest in order to forecast the travel demand in the future as well as the socio-demographic evolution

[(1)] `http://opendatacommons.org/licenses/odbl/`

Attribute	Values
Gender	male; female
Age class	0-5; 6-17; 18-39; 40-59; 60+
Age	an integer in the range [0,110]
Socio-professional status	student; active; inactive
Education level	primary; high school; higher education; none
Driving license ownership	yes; no
Activity chain	a sequence of base activity

Table 1.1 – Individuals' characteristics.

Attribute	Values
Type	single man alone single woman alone single man with children (and possibly other adults) single woman with children (and possibly other adults) couple without children (and possibly other adults) couple with children (and possibly other adults)
Children	0 to 5
Other adults	0 to 2 (mate not included)

Table 1.2 – Households' characteristics.

Figure 1.1 – VirtualBelgium environment: the 589 Belgian municipalities.

of Belgium. For instance, an individual agent should get older, give birth to new agents, die, move out, find a mate, divorce, *etc.*

Consequently VirtualBelgium contains two complementary simulation modules' sets, namely the traffic and the socio-demographic simulators as illustrated in Figure 1.2. First the initial agents are generated, then the traffic simulator module is executed. The socio-demographic evolution module can then be applied to the agents to forecast a future population for which the travel demand could be estimated by applying again the traffic simulator. Note that the temporal resolution depends on the module: one tick of the traffic simulation corresponds to one day in the agents' life while it corresponds to one year for the evolution module.

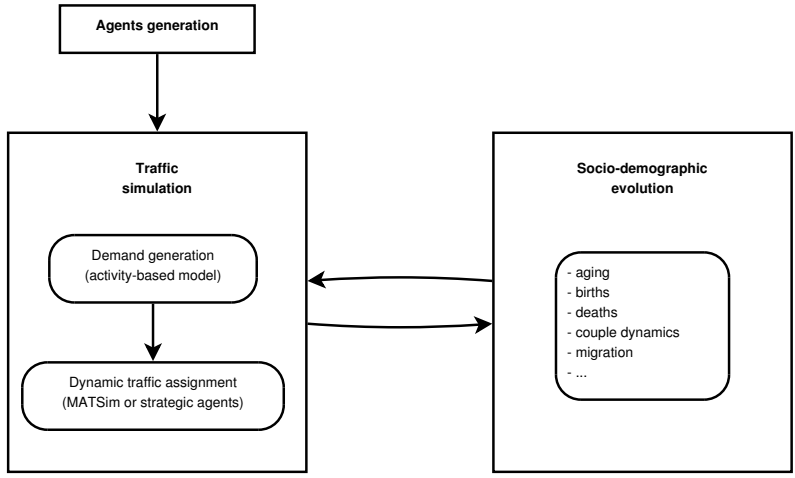

Figure 1.2 – VirtualBelgium modules.

The traffic demand generation and the dynamic traffic assignment are detailed respectively in Chapter 3 and 4 and their required inputs are listed in Appendix A. Since the scope of this thesis is focused on mobility behaviours, the evolution processes will not be discussed in this document.

1.4 VirtualBelgium's platform

The implementation of VirtualBelgium platform[(2)] has been achieved using the standard C++ programming language and relies on the Repast for Hight Performance Computing (HPC) 2.0 framework developed by Collier and North (2012) at the Argonne National Laboratory (USA).

Repast HPC is an agent-based modelling system (ABMS) intended for large-scale distributed computing platforms. This ABMS main characteristics and how it compares to related works are summarized next (Dubitzki et al., 2012):

[(2)] representing more than 15,000 lines of code in its current 1.0 version

- very large-sized model runs are supported, a feature that Drone (Koehler and Tivnan, 2005) is missing;
- the agents are based on standard object-oriented programming techniques, a more conventional approach than the finite state machines used in the FLAME-II environment (Chin et al., 2012);
- the parallelization of more than just spatial simulations as in the SWAGES platform (Scheutz et al., 2006) is possible;
- the framework relies on the Message Passing Interface, which is better suited to massively parallel machines than the OpenMP-based approaches (Massaioli et al., 2005). Indeed the latter assume a shared-memory architecture not necessarily present across all the computing nodes of a cluster;
- and it provides more flexible programming model than the ones based on graphical processing units explored by (Lysenko and D'Souza, 2008).

The developed platform runs both on regular workstation or high performance clusters running Linux systems. Submission scripts for the latter are provided with the project, for clusters running either the SLURM[(3)] or Oracle Grid Engine[(4)] resources manager. The installation and execution steps are detailed in Appendix A.

VirtualBelgium is hosted on the SourceForge platform for open-source software and can be downloaded from `http://virtualbelgium.sourceforge.net`.

1.5 Overview of VirtualBelgium's structure

An overview of VirtualBelgium's structure, which follows the standard of agent-based programming approach (van Dam et al., 2012), is illustrated on the class diagram of Figure 1.3.

The agents (Individual and Household classes), their actions and the interactions amongst them are ruled by a scheduler (belonging to the Model class). These actions take place on the Belgian road network (the Network class).

In addition to the previous classes, two singleton objects take part in the simulation. A singleton is a design pattern that restricts an object to have only one instance of itself. This is particularly useful when exactly one instance of object is required to coordinate actions and to be accessed across the system. This design patter is naturally suited for the following classes:

- the *Data* class responsible for feeding the different modules with the required inputs;

(3) `http://slurm.schedmd.com`
(4) `http://www.oracle.com`

- the *RandomGenerator* class providing various fast pseudo-random number generators. An implementation of our own generators was necessary to insure the repeatability of the simulations on different computers. Indeed there is no standard implementation for the built-in C++ random generators (Press, Teukolsky, Vetterling and Flannery, 2007) which may result in different generated pseudo-random sequences depending on the implementation. The furnished generators are detailed in Table 1.6.

The distributed computing is then performed as follows:

- the agents are uniformly distributed across the processes and added to their respective Model's SharedContext;
- each process then executes iteratively the desired modules of the simulation;
- at the end of each step, the inter-processes communications and synchronisations are handled by Repast HPC.

The documentation of every class of the simulator is provided on-line at the address `http://virtualbelgium.sourceforge.net/doc/index.html`.

1.6 Outputs and MATSim compatibility

VirtualBelgium's traffic demand simulation described in Chapter 3 generates several outputs[(5)] such as (interactive) origin-destination matrices, (animated) maps representing the number of activities performed by the agents in every municipality, *etc.* We refer the reader to the third chapter for illustrations of these outputs.

One of the main outputs consists of a XML file describing the agenda of every agent within the simulation. An example illustrating such an agenda can be found in Section A.7. This generated XML output is compatible with the open-source MATSim traffic micro-simulator currently developed jointly at TU Berlin and ETH Zürich (Balmer et al., 2009). Consequently this well-known and validated framework can be coupled with VirtualBelgium in order to perform dynamic traffic assignment of the generated demand.

This traffic micro-simulation framework has been retained amongst the existing agent-based ones[(6)] for its open-source nature, active maintenance, support from the MATSim community, and successful applications in several location including Switzerland (Meister et al., 2010), Tel Aviv (Bekhor et al., 2012), the Netherlands (Horni et al., 2009) and Toronto (Gao et al., 2010*b*).

[(5)] detailed in Appendix A

[(6)] see the introduction of Chapter 4 for a brief literature review

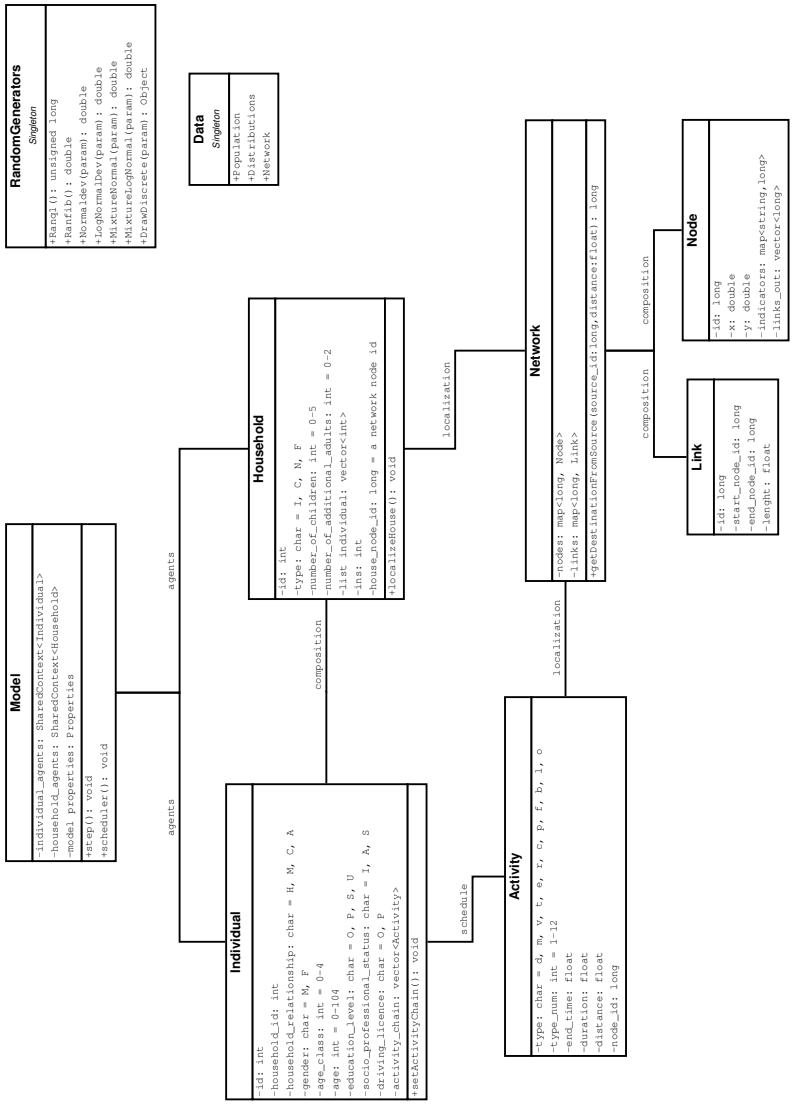

Figure 1.3 – Class diagram.

				Shape parameters	
Generator	**Distribution**	**C++ type**	**Range**	**Location**	**Scale**
Ranq1	Uniform	unsigned long	$[1, 2^{64} - 1]$	/	/
Ranfib	Uniform	double	$]0, 1]$	/	/
Normaldev	Normal	double	$\mathbb{R}$	$\mu \in \mathbb{R}$	$\sigma \in \mathbb{R}$
LogNormadev	Log-Normal	double	$\mathbb{R}_0^+$	$\mu \in \mathbb{R}$	$\sigma \in \mathbb{R}$
MixtureNormal	Mixture of k independent Normal	double	$\mathbb{R}$	$\boldsymbol{\mu} \in R^k$	$\Sigma \in \mathbb{R}^{k \times k}$
MixtureLogNormal	Mixture of k independent Log-Normal	double	$\mathbb{R}_0^+$	$\boldsymbol{\mu} \in \mathbb{R}^k$	$\Sigma \in \mathbb{R}^{k \times k}$
DrawDiscrete	Empirical discrete distribution	any	/	/	/

Table 1.3 – Pseudo-random number generators. Ranq1, Ranfib and Normaldev are inspired from Press et al. (2007).

Chapter 2

Synthetic population generation

Contents

2.1 Introduction

Micro-simulations, such as the activity-based travel demand model developed within VirtualBelgium for transport demand forecasting, usually involve a large number of agents. It then may be impossible or too expensive to obtain a fully disaggregated data set describing the agents of interest. Moreover, if such a data set were available, its use may also be problematic in some countries due to stringent privacy laws. A way to address these issues is to construct an artificial population starting from known data about the true one. Consequently synthetic population generation has recently received considerable attention in the literature (Müller and Axhausen (2011) presents a good overview of the techniques available in 2011). As it is obvious that the representativeness of the synthetic population is critical for the simulation accuracy, a synthetic population generator should therefore produce an artificial population approximating the correlation structure of the true population as accurately as possible.

Techniques for synthetic population generation typically belong to either the Synthetic Reconstruction (SR) techniques or the Combinatorial Optimization (CO) methods. The SR methods generate a synthetic population given joint-distributions of the population's attributes, generally using a sample of the population and the iterative proportional fitting procedure (IPFP) to generate the desired joint-distributions (see Wilson and Pownall, 1976 and Beckman et al., 1996). The CO category is far less common. The CO methods divide the area of interest in mutually exclusive zones for which a set of marginal distributions of the desired attributes is available. Then a sub-set of a sample taken over the whole population is fitted to the given set of margins for each zones. We refer the reader to Voas and Williamson (2001) and Huang and Williamson (2002) for a formal and complete description of these latter methods.

However, both SR and CO approaches usually make strong assumptions on the data used in the process, and it is not always possible to ensure that they can be satisfied in practice. In particular, this caused significant difficulties in the generation of a synthetic population for Belgium. These difficulties motivates the research presented here, where a new type of SR generator is developed, obviating these data-related issues.

This chapter, based on Barthelemy and Toint (2013), is organized as follows. In Section 2.2, we first present the standard approach for building a synthetic population, from which the other Synthetic Reconstruction techniques are derived. Section 2.3 then describes an alternative method, belonging to the SR family, obviating the limitations of the conventional generation methods. We next present in Section 2.4 the results of this new methodology applied to the generation of a synthetic population for Belgium. Section 2.5 then compares the new generator with an IPFP-based methodology. Concluding remarks are finally discussed in Section 2.6.

2.2 The standard approach

To date, the standard approach for building synthetic populations is based on the method developed by Beckman et al. (1996), whose main idea consists in merging aggregate data from a source covering the whole population with disaggregated data from a sample in order to get a disaggregated data set for the population of interest. Typically the aggregate data set is extracted from an existing census and the disaggregated data set is drawn from a survey over a sample of the population. The aggregate data consists of a set of marginal distributions for the characteristics of interest of the true population: we refer to these distributions and variables as the target and control variables. The disaggregated data provides full information about the attributes of interest, but only for a sample of agents, and is referred to as the seed.

The population synthesis procedure usually starts with identifying the relevant (categorical) socio-demographic variables of the agents. Assuming that there are n attributes of interest in the seed and denoting by $V = \{v_1, v_2, ..., v_n\}$ the vector of variables representing these attributes, each combination of values of v_i therefore defines a socio-demographic group. The synthetic population is then generated by a 2-steps procedure:

1. Starting from the seed, estimate the $k-$way joint-distribution of the true population, where $k \leq n$ is the number of control variables, such that the resulting distribution is consistent with the marginal distributions (margins) of the target and preserves the correlation structure of the seed.

2. Select agents from the sample and copy them to the synthetic population in a proportion derived from the distribution computed in the previous step.

These steps are discussed in the next two subsections, followed by a description of the limitations of this first approach and the proposed improvements obviating these limitations.

2.2.1 Estimating the attributes' joint-distribution using IPFP

The most popular way to estimate a $k-$way joint-distribution table based on known marginal distributions and a sample is the well-known iterative proportional fitting procedure (IPFP) originally described by Deming and Stephan (1940). This procedure is detailed below for $k = 2$, but can easily be extended to higher dimensions.

Assume that a 2-way contingency table is built from the seed with initial components $\pi_{ij} \in \mathbb{R}^+$ were i and j respectively correspond to the level of the first and the second variable. These π_{ij} correspond to the number of agents in the sample for each combination of levels. Assume also that desired marginal

distributions $\{x_{i\bullet}, x_{\bullet j}\}$ (the target) are known $\forall i, j$. The IPFP then iteratively updates the cells' values depending on the marginal distributions of the target until the margins of the computed table match the target's ones, *i.e.* $\pi^*_{i\bullet} = x_{i\bullet}$ and $\pi^*_{\bullet j} = x_{\bullet j}$ where the π^*_{ij} are the component values at the last iteration. The adjustments at iteration l are computed by the equations

$$\pi^{l'}_{ij} = \pi^{l-1}_{ij} . \frac{x_{\bullet j}}{\pi^{l-1}_{\bullet j}} \qquad \forall i, j; \tag{2.1}$$

$$\pi^{l}_{ij} = \pi^{l'}_{ij} . \frac{x_{i\bullet}}{\pi^{l'}_{i\bullet}} \qquad \forall i, j. \tag{2.2}$$

In order to produce an accurate estimate of the true distribution, the procedure ideally requires an initial representative sample of the true population for building the initial multiway table (even if, technically, a multiway table of ones can be used as a starting point of the procedure). This requirement is important since Mosteller (1968) pointed out that the procedure preserves the interaction structure of the sample as defined by the odd ratios

$$\frac{\pi_{ij} . \pi_{hk}}{\pi_{ik} . \pi_{hj}} = \frac{\pi^l_{ij} . \pi^l_{hk}}{\pi^l_{ik} . \pi^l_{hj}} \tag{2.3}$$

at each iteration l, were $i \neq h$ and $j \neq k$ stand for different levels for each variable respectively. Moreover, according to Ireland and Kullback (1968), the IPFP also produces the π^*_{ij} minimizing the discrimination information, also known as the relative entropy or the Kullback-Leibler divergence, defined by

$$\sum_i \sum_j \pi^*_{ij} \ln \left(\frac{\pi^*_{ij}}{\pi_{ij}} \right). \tag{2.4}$$

Finally, Little and Wu (1991) showed that IPFP results in a maximum likelihood estimator for the RAKE model which was judged "the best overall choice ... in the absence of knowledge of the form of the selection model [for fitting to known marginals]".

2.2.2 Generating the synthetic population

Once the expected numbers of agents in every socio-demographic group are estimated, each sampled agent is associated with a probability of being included in the synthetic population. This probability typically depends on the agent's sampling weight and the expected number of similar agents in the true population. Based on these probabilities, the approach of Beckman et al. (1996) randomly draws agents from the sample using a Monte-Carlo procedure until the expected number of agents is reached for each socio-demographic group. When a sampled agent is drawn, then all its attributes, including the uncontrolled ones, are pasted to a new synthetic agent who is added to the synthetic population.

2.2.3 Limitations and improvements of the approach

Recent mobility surveys such as EGT (Direction Régionale de l'Équipement d'Île-de-France, 2004), MOBEL (Hubert and Toint, 2002) or NTS (Avery, 2011) suggest that the travel behaviour of an individual is significantly influenced by the type and composition of his/her household. This points to a first limitation of the conventional approach: it is very unlikely that analysts have access to a single dataset detailing the joint-distribution of individuals' and households' attributes simultaneously. Since the estimation step of the algorithm described in Section 2.2.1 is designed to deal with a single contingency table, the conventional approach can consequently account either for individual-level or for household-level control variables but not for both. In other words this process results in a synthetic population where either the households' or individuals' joint-distributions match the desired ones but not both. Note that households' distributions accuracy has often been preferred (Ye et al., 2009).

This strong limitation led several authors to propose interesting improvements to this basic algorithm. Guo and Bhat (2007) designed a method to overcome this problem by simultaneously controlling the individual- and household-level variables. Their algorithm generates a population where the household-level distributions are close to those estimated using the IPFP, while simultaneously improving the fit of person-level distributions. Arentze et al. (2007) propose another method using relation matrices to convert distributions of individuals to distributions of households, such that marginal distributions can be controlled at the person level as well. Ye et al. (2009) further build on these contributions and suggest a practical heuristic approach called Iterative Proportional Updating (IPU), based on adjusting households' weights such that both household- and individual-level distributions can be matched as closely as possible. Control for households and individuals relationship, improvements or alternative to the standard approaches are also investigated in Auld et al. (2010), Pritchard and Miller (2009), Srinivasan et al. (2008) and Huynh et al. (2013).

However, these improved approaches remain based on the IPFP (or the IPU) and thus rely on the same assumptions on data quality, *i.e.* that the aggregate data of the target is consistent in the sense that margins extracted from available but different joint-distributions are equal. This is critical for practical convergence of IPFP. They also assume that a significant sample of the population of interest is available at the desired level of disaggregation, from which synthetic agents can be extracted and duplicated. For example if a class of agents is not represented in the seed then this particular class will remain unpopulated in the final synthetic population. This could also be cured by introducing small initial values in the unpopulated classes but this approach remains unsatisfactory as it introduces unwanted bias.

These two strong requirements unfortunately limit the applicability of the IPFP in real situations, such as the generation of a synthetic population for Belgium at the municipality level. Indeed these requirements could not be met

in this particular case. Firstly, a representative sample at the municipality level (which is the desired spatial disaggregation level) is not available, and, even if it was, the privacy issue would remain because the IPFP repeats the observations of the sample as many times as necessary. Gargulio et al. (2010) have proposed a sample-free generator to overcome this issue whose performance have been favourably compared to a sample-based generator (Lenormand and Deffuant, 2013). A second problem is that all necessary informations, *i.e.* distributions, are not available from a single source (which would hopefully guarantee consistency), but have to be extracted from different datasets, typically produced by different institutions and/or using different protocols or data cleaning mechanisms. This results in significant differences between margins, as illustrated in Table 2.1 (extracted from Cornélis et al., 2005) for the Charleroi district.

Joint-distribution	**Data Source**	**Margins**	**Prop.**
municipality × gender × age	GéDAP, 2001	405.491	1,00
municipality × household type	GéDAP, 2001	380.653	0,94
municipality × education level	GéDAP, 2001	426.372	1,05
municipality × activity status	GéDAP, 2001	396.594	0,97
district × household type × age	INS, 2001	357.884	0,88
district × education level	INS, 2001	398.582	0,98

Table 2.1 – Inconsistencies between margins extracted from different sources.

In this table, the total number of inhabitants in the district is compared among the data files used in the population synthesis. If one takes (for instance) the first dataset as a reference, one immediately notices inconsistencies between the different estimations with differences up to 12%, irrespective of the data source (see last column of Table 2.1). These inconsistencies prevent the IPFP process to converge. This could possibly be cured by considering the frequencies rather than the number of agents themselves, but the issue of the missing sample nevertheless remains. These difficulties motivate our proposal for an alternative population synthesis tool which would not suffer from the lack of a representative sample at the most disaggregated level and/or from (moderate) inconsistencies between different data sources. This is the object of Section 3.

2.3 A new population synthesis technique

We start the presentation of our proposal by outlining its main steps before the more formal description.

Our general philosophy is to construct individuals and households by *drawing their characteristics or members at random within the relevant distribution at the most disaggregate level available, while maintaining known correlations as well as possible.* The algorithm implementing this principle, and illustrated in Figure 2.1, consists of a 3-steps procedure for each spatial aggregation unit:

1. a pool of individuals is generated, which we denote by Ind;
2. the households' joint-distribution is estimated and stored in the contingency table Hh;
3. the synthetic households are constructed by randomly drawing individuals from the individuals' pool Ind. This is achieved while preserving the distribution computed in the second step. Once a household has been built, it is added to the synthetic population.

We now provide detailed information on each of these successive steps.

2.3.1 Step 1: Generating the pool of individuals

The first step aims at building the Ind pool of synthetic individuals for the area of interest, by generating them one by one. In our method, each individual is characterized by a vector of attributes $V = (V_1, \ldots, V_n)$, whose components may take a discrete set of values. We denote by v_i the value taken by the characteristic V_i for a particular individual. We would like to draw each v_i from known empirical distributions derived from available data. However, not every distribution for V_i is known at the most disaggregate level, and we thus face a hierarchy of levels. Our first step is then to merge the various distributions available at the same disaggregation level using the IPFP technique (Frick and Axhausen (2004) and Guo and Bhat (2007)), possibly substituting less reliable values by their frequencies to handle inconsistent margins. This results in a set of distributions V^k, where k denotes the level of disaggregation (in our case, municipality, district, nation). In accordance to the general principle stated above, our idea is then to use, for each such characteristic, the most disaggregate level available.

Specifically, a table V^0 corresponding to the numbers of individuals with the attributes $(v_1^0, \ldots, v_{n_0}^0)$ is first constructed from the most disaggregated data available (at municipality level in our case). The missing attributes for each individual in this table are then determined by finding the most disaggregate level at which a joint distribution for the missing attribute and some already known characteristic of the considered individual is available. The first of these is then determined by a random draw in this (conditional) distribution. Once all characteristics of an individual are defined, the pool Ind is updated.

Since some of the individuals' characteristics are determined by draws from distributions at aggregate levels, the margins extracted from the pool Ind for these particular characteristics may be inconsistent with the known true ones. For each attribute, a correction is then made to the agents of Ind to make it consistent with their respective margins at the level 0. This correction is computed by suitably shifting the attribute's value of certain number of individuals, determined by the number of individuals with problematic attribute's values and the known distribution of this attribute. Only shifts between two contiguous modality are allowed. For instance consider an attribute U whose

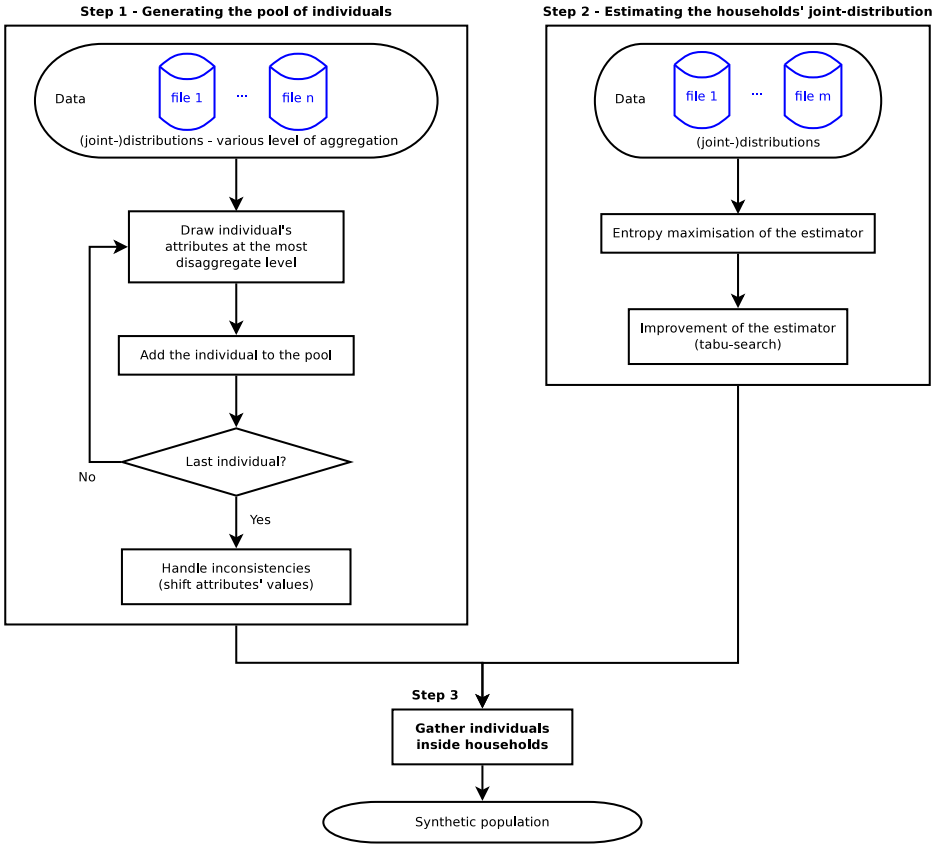

Figure 2.1 – Synthetic population generator.

modalities are u_1, u_2, u_3 and u_4, where

$$u_1 \leq u_2 \leq u_3 \leq u_4. \tag{3.5}$$

If an individual is initially characterised by u_2, then the shift is either u_1 or u_3. Note that these shifts can only be applied to numerical or ordinal variables.

2.3.2 Step 2: Estimating the households' joint-distribution

We now consider the second step of our population synthesis procedure. Denote by $W = (W_1, \ldots, W_m)$ the vector the of household-related attributes and by w_j the value taken by a particular households for the j^{th} such attribute. Now that a pool of individuals has been built, the next step is to find an estimator of the households' type contingency table, denoted by Hh, given data provided by several different sources. Each cell of Hh thus corresponds to the number of particular household of a type specified by a combination of the w_j's (which we call a household type). This problem is solved in two steps: a maximum entropy estimate of Hh is first generated and subsequently improved by using a tabu-search optimization process.

2.3.2.1 Entropy maximisation of the estimator

In our algorithm, the initial estimation of Hh is obtained as the solution of an optimization problem, where the entropy is maximized under the (linear) constraints implied by the known margins on households' types. This approach has the advantages of producing a more reasonably spread-out distribution amongst all types while keeping the constraints satisfied than would be produced by a least-squares formulation, say. The entropy maximization approach is introduced here in an intuitive way inspired by Bierlaire (1991) and Ortúzar and Willumsen (2001). For a more formal description, see Wilson (1974).

Consider a system consisting of a large number of distinct elements. A full description of such a system requires the complete specification of each micro-state of the system which involves in our case completely identifying each household. At this stage, we are however, interested in a more aggregate level called the meso-state, corresponding to the households' distribution Hh. Typically one meso-state can be associated with different micro-states. For instance if two household heads with similar attributes are exchanged, then the meso-state is unchanged but the associated micro-states are different. Finally, the last and highest level of aggregation called the macro-state is the available data on the system as a whole.

The basic idea of the method is to accept that, unless we have information on the contrary, all micro-states consistent with the macro-state are equally likely. This consistency is enforced, in our approach, by imposing equality constraints given by the macro-state. If $x = (x_1, \ldots, x_p)$ is the vector of unknown cells of Hh, Wilson (1970) showed that the number of micro-states $E(Hh)$ associated

with the meso-state Hh is given by

$$E(Hh) = \frac{(\sum_i x_i)!}{\prod_i x_i!}. \tag{3.6}$$

The function $E(\cdot)$ is called the entropy function. As it is assumed that all micro-states are equally likely, the meso-state corresponding to the largest number of micro-states (and thus the most likely) is that maximizing (3.6). Using the natural logarithm and Stirling's short approximation (Dwight, 1961 and Kreyszig, 1972), the corresponding objective function of this problem can then be approximated by

$$\min_x \sum_i x_i \ln(x_i) - x_i \tag{3.7}$$

under the constraints on households types given by the macro-state.

Unfortunately, due to the inconsistent nature of the available data, as exposed in Section 2.2.3, the constraints of this optimization problem are also formally inconsistent. Our approach is then to impose only a subset Ω of them corresponding to the data of highest quality as strict constraints, the others being then incorporated in the objective function in a form penalizing their violation. Each of these latter constraints p is affected with a weight defined by $n_p\sigma$ where σ is a penalization parameter and n_p is the number of households involved in p.

Note that the problem constraints are all linear, and can therefore be represented in matrix form by the system $Ax = b$, where A contains the coefficients of the variables and b the independent terms. Denoting by A_σ and b_σ the matrix and the vector derived from the subset of the scaled inconsistent constraints, the new objective function can now be formulated as

$$(EN) \qquad \min_x \|A_\sigma x - b_\sigma\|_2^2 + \sum_i x_i \ln(x_i) - x_i \tag{3.8}$$

and the minimization is then carried out under the constraints in the set Ω only. This optimization problem is solved using an augmented Lagrangian algorithm, as implemented in the (freely available) LANCELOT package (Conn et al., 1992 and Gould et al., 2003). In general the solution of this optimization problem yields a non-integer solution, which is unsuitable for representing households' numbers. The solution's components of this optimisation problem are then rounded and the value.

$$f_{EN}(\hat{x}) = \sum_i w_i|\hat{c}_i - c_i| + \sum_i \hat{x}_i \ln(\hat{x}_i) - \hat{x}_i \tag{3.9}$$

is computed, where $\hat{x}$, $\hat{c}_i$, c_i and w_i denote the rounded solution of (EN), the computed and the desired value of the i^{th} constraint and the associated weight depending on the quality of the associated data source, respectively. The latter are admittedly somewhat arbitrary: we have chosen to penalize violation of

consistent constraints ten times more than those associated with inconsistent ones, but the results seem relatively insensitive to this choice. The value of f_{EN} can be seen as a performance measure describing how well the rounded integer solution fits the whole set of initial constraints. We then loop over a set of values for the penalization parameter σ, and the best rounded solution x^* associated with the lowest value of f_{EN} is determined. This solution is finally used as the starting point of a combinatorial optimization problem using a tabu-search algorithm in order to get a final estimation of Hh. Details on this process are provided in the next subsection.

2.3.2.2 Improvement of the estimator using tabu-search

Tabu-search is a local-search meta-heuristic originally proposed by Glover (1986), which can be used for solving combinatorial optimization problems. This procedure iteratively moves from one solution x to a solution $x' \in \mathcal{N}(x)$, a neighbourhood of x containing a list of candidate solutions, until a stopping criterion (such as a given number of iterations N) has been reached. In order to avoid cycling, the neighbourhood $\mathcal{N}(x)$ is modified to exclude some solutions encountered in previous iterations (these solutions constitute the *tabu list*). For a complete description of this optimization technique, we refer the reader to Glover (1989), Glover (1990) and Glover and Laguna (1997).

The chosen tabu list is a list T of size n, which contains the solutions visited in the last n iterations. If we denote by x^i the candidate solution at iteration $i > 0$, x^0 being x^* *i.e.* the rounded solution computed above, $\mathcal{N}(x^i)$ is then defined as follow:

$$\mathcal{N}(x^i) = \left\{ x^i_{j\pm} = (x_1^{i-1}, \ldots, x_j^{i-1} \pm 1, \ldots, x_p^{i-1}) \;\mid\; j = 1, \ldots, p \right\},$$

where the notation $x_j^{i-1} \pm 1$ stands for two variations of the j-th component around its value x_j^{i-1}. The following steps are then executed iteratively N times:

1. define a new candidate by randomly drawing $x^i \in \mathcal{N}(x^{i-1})$ such that $x^i \notin T$;
2. if $f_{EN}(x^i) < f_{EN}(x^*)$ then $x^* = x^i$;
3. replace the oldest component of T by x^i and go back to Step 1.

This procedure results in an updated and improved estimate x^* of Hh. Note that the quality of the improvement depends on the size of the tabu list and the number of iterations allowed. These parameters must therefore be chosen to obtain a reasonable trade-off between computing cost and quality of the estimate. However, the impact of varying these parameters appears to be small in our application.

2.3.3 Step 3: Households' generation

Individuals' and households' distributions being estimated, the last step of our generator consists in gathering individuals into households by randomly drawing households' constituent members. We proceed in two successive stages: the first is to select a household type and the second to draw the individuals to form a household of this type.

The selection of the household type is performed in order to keep the distribution of already completed households statistically close to the estimated one. The goal is achieved by choosing the type of the next household to assemble such that the distribution Hh' of the already generated households (including the household being built) minimizes the observed χ^2 distance between the Hh and Hh', which is given by

$$d_{\chi^2}(Hh', Hh) = \sum_{i}^{p} \frac{(x'_i - x_i)^2}{x_i^2}.$$

This minimization is extremely simple because the number of household types is very limited. Once the household type is selected, household's members are generated as follows: a household head is first drawn from the pool of individuals Ind, and then, depending on the household's type, additional individuals are also drawn from the pool if relevant. All these draws from Ind are made without replacement.

We now provide some detail on this last drawing process. If we assume that a household is made of a head and possibly a mate, children and additional adults, the construction starts with the selection of its head. Depending on the household type, the head's attributes are either obtained directly (for instance for an isolated man) or randomly drawn according to known joint-distributions, *e.g.*

- household type × head's gender × head's age;
- household type × municipality type × head's age × head's activity status;
- household type × municipality type × head's age × head's education level.

More formally, this selection procedure is organized in 3 steps:

1. Determine the desired attributes' values (*i.e.* the v_i's) for the household head:
    - some can be derived directly from the current household type;
    - the remaining missing attributes are either randomly drawn according to known distributions or, if different values are feasible and equally likely for V_i, determined in order to minimize the χ^2 distance between the generated and estimated distributions.

2. Add the head to the household being generated:
    - if the corresponding individual's class is still populated in the individuals' pool, extract an individual from this class and make it the household's head;
    - else find a suitable household head by random search in the constituents members of the previously generated households. This last individual is then replaced with an appropriate one randomly drawn in the pool of the remaining individuals. If the generator fails to find a head, then the generation is ended.
3. The estimated and generated contingency tables are updated according to the actions performed in Step 2.

Depending on the household type, the generator may pursue the construction of the current household by selecting a head's partner, children and additional adults. The corresponding selection procedures are similar to the head's one, with the only exception that individuals' characteristics may no longer be determined by the household type only, but are randomly drawn according to known distributions on couple formation such as

- household type × head's gender × head's age × mate's age;
- household type × head's gender × head's education level × mate's education level;

or by predefined rules (a child must be younger than his/her parents).

The household generation for the current municipality terminates if all households have been constructed, or the generator fails to find a household member, *e.g.* if the pool of individuals is empty or if it is impossible to find a suitable individual in the previously generated households.

When the procedure stops after exhausting either the pool of individuals or the pool of households, inconsistencies of two types may remain in the generated population: in the first case the final number of households is smaller than anticipated, while the final number of individuals is smaller than estimated in the second case.

Note that the type of each household to be generated results from an optimization process. If this selection has been random, we would have to generate several populations in order to select the best one amongst them (in the sense of statistical similarity), which can lead to heavy additional computational costs.

2.4 Generating a Belgian synthetic population

2.4.1 The application and data sources

The procedure outlined in the previous section has been used to generate a synthetic population of 10,637,107 individuals gathered in 4,334,281 households for

the 589 municipalities of Belgium in 2001. The municipalities (LAU-2 level) themselves belong to 43 districts (NUTS-3 level) containing between 2 and 35 municipalities each. Table 2.2 presents basic statistics on these municipalities. The individuals' and households' attributes are respectively described in Tables 2.3 and 2.4. The maps illustrated in Figures 2.2 and 2.3 represent respectively the proportions' spatial distribution of the 60+ years old individuals and working individuals in the synthetic population.

	Min	**Max**	**Mean**
Individuals	85	461,115	18,059.6
Households	35	212,707	7,358.9

Table 2.2 – Basic statistics for municipalities

Attribute	**Values**
Gender	male; female
Age class	0-5; 6-17; 18-39; 40-59; 60+
Activity	student; active; inactive
Education level	primary; high school; higher education; none
Driving license ownership	yes; no

Table 2.3 – Individuals' characteristics

Attribute	**Values**
Type	single man alone single woman alone single man with children (and other adults) single woman with children (and other adults) couple without children (and other adults) couple with children (and other adults)
Number of children	0 to 5
Number of other adults	0 to 2 (mate not included)

Table 2.4 – Households' characteristics.

Data available at the municipality or district aggregation levels is provided from the following sources:

- *Directorate-general Statistics and Economic information* of the Belgian Federal Government (2001);
- *Service public fédéral Mobilité et Transports* of the Belgian Federal Government (2000);

- *Groupe d'étude de démographie appliquée* (GéDAP) centre of the Université catholique de Louvain (2001);
- the *MOBEL* mobility survey (Hubert and Toint (2002)).

The generator has been implemented in a single threaded software written in the *Perl* 5.10 programming language and executed on a desktop computer running with an Intel(R) Core(TM) 2 Duo @ 3Ghz E6850 CPU and 3Gb of RAM under a 32 bits Linux environment. The generation process uses a TABU list of length 1000 and takes around 16 hours and 30 minutes to handle all the 589 municipalities.

2.4.2 Verification of the household generation procedure

2.4.2.1 Absolute percentage difference

Having generated a synthetic population, one is then faced with the question of estimating its quality. As in Guo and Bhat (2007), one possible performance measure to assess the generator accuracy is the absolute percentage difference (APD), also known as the relative error, between the estimated contingency tables computed in the first steps (Steps 1 and 2) of the generator and the corresponding ones resulting from the household generation step (Step 3). This measure is calculated for a particular cell $(u_1, \ldots, u_p)$ as follows:

$$APD_{T,T'}(u_1, \ldots, u_p) = \left| \frac{T'[u_1] \ldots [u_p] - T[u_1] \ldots [u_p]}{T[u_1] \ldots [u_p]} \right|$$

where T and T' denote respectively the estimated (Steps 1 and 2) and the generated (Step 3) tables. The lower the APD, the better the generated table fits the estimated one. Results are reported in Table 2.5.

	Estimated	**Generated**	**Difference**	***APD***
Individuals	10,637,107	10,635,691	1,416	< 0.001
Households	4,334,281	4,333,448	833	< 0.001

Table 2.5 – Generated agents.

First note that the procedure was able to generate 10,635,695 individuals gathered in 4,333,425 households, meaning that it could build a synthetic population where the numbers of households and individuals are very close to the estimated ones and differs less than 0.1% for the number of agents. This is highly encouraging.

Table 2.6 presents some basic statistics (minimum, maximum, standard deviation and mean) on the average APD values ($AAPD$, also known in the literature as the mean absolute percentage error) of the cells of the generated distributions computed across all the municipalities. As one can easily see, all these statistics also seem to indicate that the generator produces an accurate

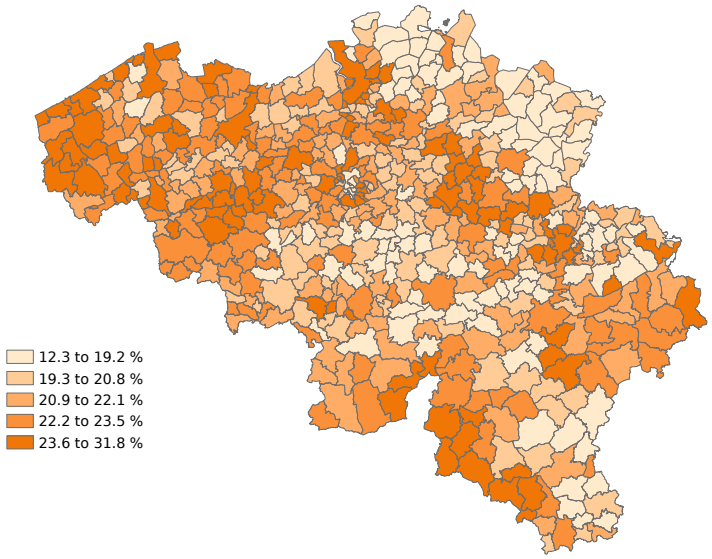

Figure 2.2 – Percentage of 60+ years old people by municipality.

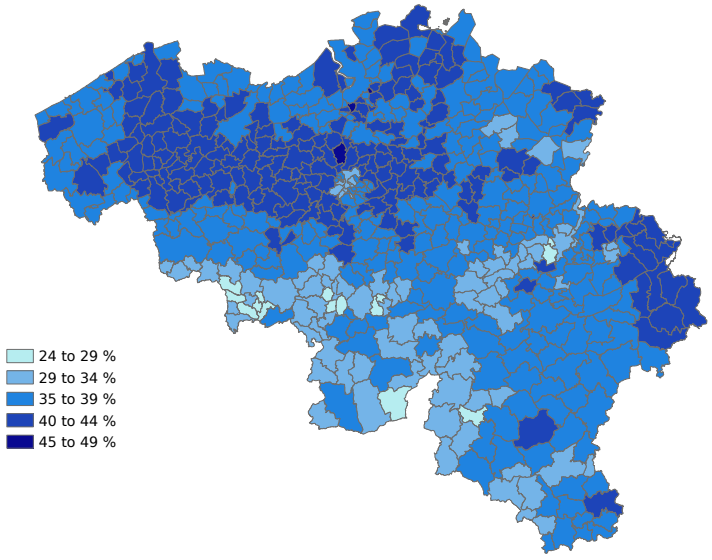

Figure 2.3 – Percentage of working individuals by municipality.

synthetic population. The maximum $AAPD$ value for Hh' is associated with the municipality of Herstappe, which contains only 85 inhabitants gathered in 35 households. Due to its small size, a small deviation from the desired Hh can easily, in this case, result in a relatively large $AAPD$ of 8.2%. Table 2.7 presents the same statistics of Table 2.6 where we have neglected this problematic municipality, showing it can be considered a statistical outlier.

Distribution	**Min**	**Max**	**Std dev**	**Mean**
Ind'	0.000	0.005	$<$ 0,001	$<$ 0,001
Hh'	0.000	0.082	0,003	$<$ 0,001

Table 2.6 – $AAPD$ statistics.

Distribution	**Min**	**Max**	**Std dev**	**Mean**
Ind'	0.000	0.005	$<$ 0,001	$<$ 0,001
Hh'	0.000	0.003	$<$ 0,001	$<$ 0,001

Table 2.7 – $AAPD$ statistics without Herstappe.

At a more disaggregate level, Figures 2.4 and 2.5 illustrate the $AAPD$s' repartition for the individuals' and households' types across the Belgian municipalities and give some evidence of the synthetic population's accuracy in terms of $AAPD$ and spatial coherence. Figures 2.6 and 2.7 give a representation of the APD's mean and the standard deviation of each individual and household type over the 589 municipalities. Again, these figures suggest that the generator produces relatively small APD on average. Moreover, these APDs are associated with small standard deviations, meaning that APD values are relatively stable across the municipalities.

The synthetic populations associated with the worst $AAPD$ values for the individuals and the households are respectively those for Erezée and Herstappe. The details of these municipalities are described in Table 2.8. They clearly indicate that, even if these entities are the less accurate ones, the generated distributions are still reasonably close to the estimated ones: in average, the APD between the estimated and generated distributions for a given individual class is less than 0.5% while it is less than 8.2% for a given household type. Additionally the generator produces a population having $<$ 0.1% less individuals and 7.9% less households than the estimated one. These results are illustrated on Figures 2.8 and 2.9 representing the number of agents generated against the number of estimated one for each class of agents. As one can easily see, the contingency tables produced by the generator fit the initial ones quite accurately, given the initial level of data inconsistencies.

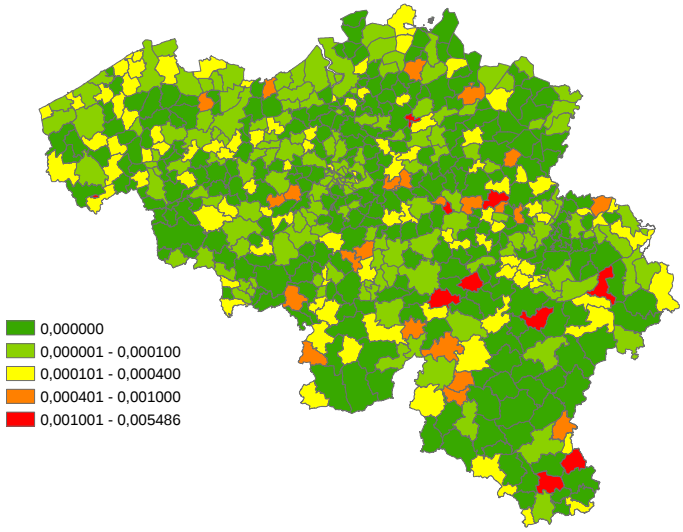

Figure 2.4 – $AAPD$s' repartition for the individual's types

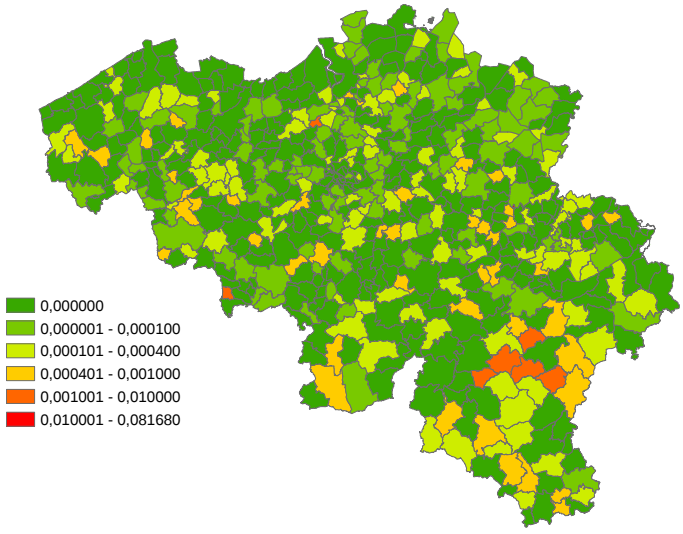

Figure 2.5 – $AAPD$s' repartition for the household's types

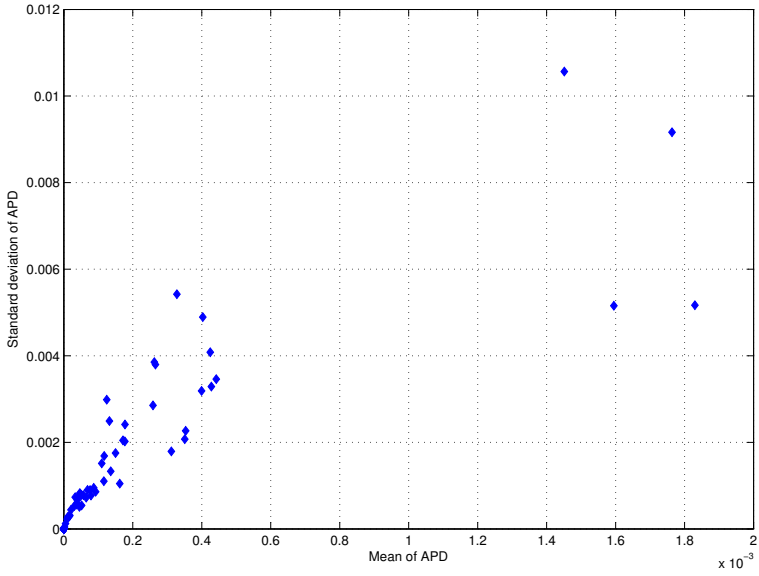

Figure 2.6 – $AAPD$s' mean and standard deviation for each individual type.

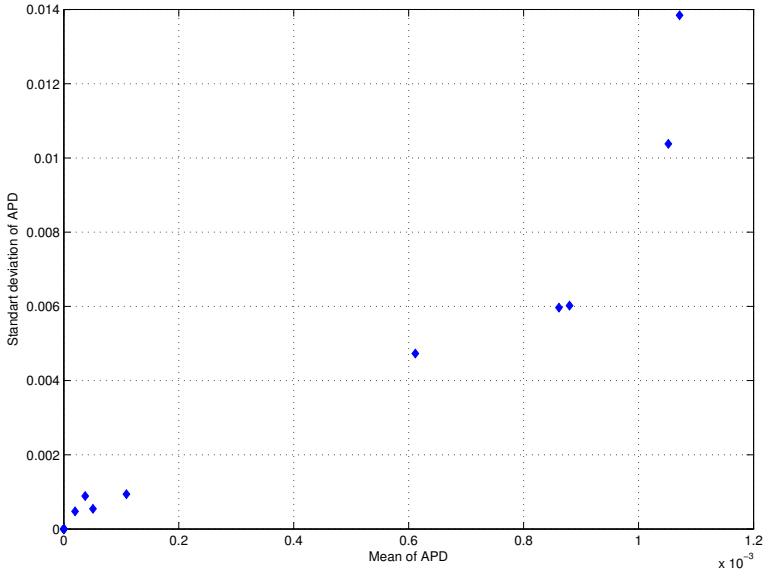

Figure 2.7 – $AAPD$s' mean and standard deviation for each household type.

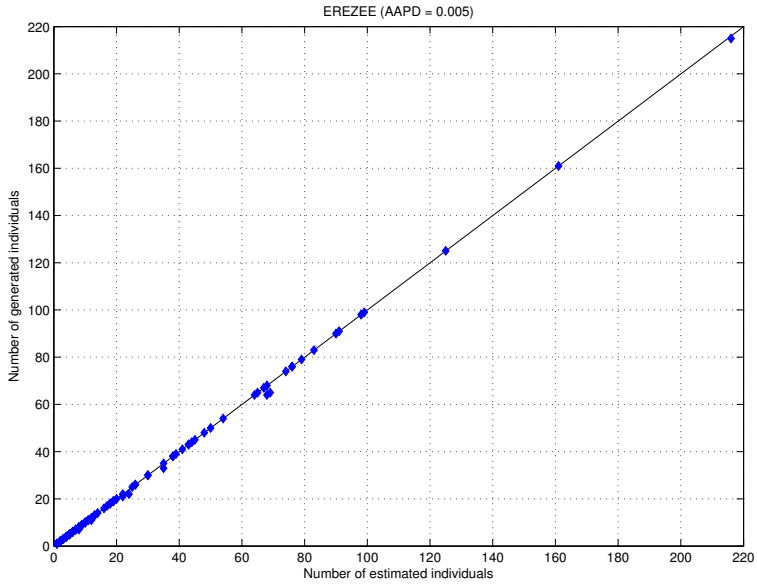

Figure 2.8 – Estimated vs. generated individuals for Erezée (the worst municipality for this type of agent).

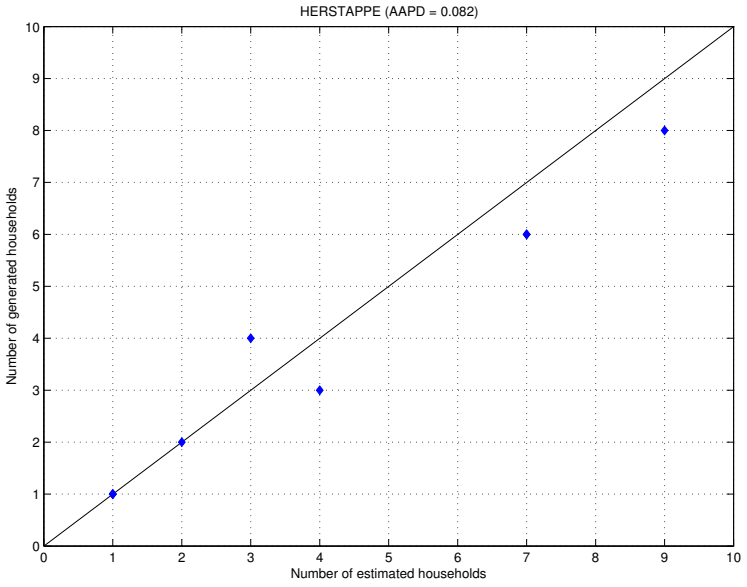

Figure 2.9 – Estimated vs. generated households for Herstappe (the worst municipality for this type of agent).

	Erezée	**Herstappe**
Distribution (D)	Ind	Hh
Agents Estimated (E)	2,885	38
Agents Generated (G)	2,869	35
Difference	16	3
$APD(E,G)$	< 0.001	0.079
$AAPD(D,D')$	0.005	0.082

Table 2.8 – Erezée and Herstappe.

2.4.2.2 Freeman-Tukey goodness-of-fit test

Finally, we evaluate the goodness-of-fit of the distributions produced by households generation procedure to the estimated ones at Steps 1 and 2. This comparison is achieved by using the Freeman-Tukey statistic defined by

$$FT(T,T') = 4\sum_i \left(\sqrt{T_i} - \sqrt{T'_i}\right)^2$$

where T and T' are respectively the estimated and generated (household or individual) distributions. This test, suggested by Voas and Williamson (2001), has the advantage over the classic Pearson χ^2 test that it allows the presence of zeros in the cells of the distributions. The FT statistic follows a χ^2 distribution with a number of degrees of freedom equal to one less than the number of cells in the compared distributions. The results of this goodness-of-fit test are highly promising as all generated distributions were statistically similar to the estimated ones both at a 95% and 90% levels of confidence.

2.5 Comparison with an IPFP-based generator

2.5.1 Data and parameters

In order to further assess the reliability and accuracy of our generator, we now compare it with an IPFP-based synthetic population generator, namely the extended IPFP generator described by Guo and Bhat (2007). The results obtained by this generator are strongly influenced by a parameter denoted by $PDTS$, whose value has been set to 0.10 for the comparison experiments. This value is recommended by Guo and Bhat (2007) and our experience also shows that it worked best with our data.

Assuming that the population generated in the previous subsection is a real population, we generate two synthetic populations by using the two generators. As the true population is known, the required data necessary for running the two generators can easily be extracted, *i.e.* on one hand a significant sample of households for each municipality and a set of margins for Guo and Bhat's generator, and, on the other hand, various joint-distributions (as specified above)

at the municipality and district level for the new generator. Since the entire true population is known, the extracted data used is clearly consistent and the IPFP-related assumptions are met. The comparison can then be done without loss of accuracy. All the individuals' and households' attributes are used as control variables for the IPFP process. Due to its small size, the municipality of Herstappe is not considered in this test.

The minimal size of the sample for each municipality is given by (Levy and Lemeshow, 1999)

$$n \geq \frac{z_{1-\alpha/2}N}{z^2 + (N-1)l^2} \tag{5.10}$$

where N is the total number of individuals of a municipality, l the margin of error and $z_{1-\alpha/2}$ is the reliability coefficient associated with a confidence interval of $(1-\alpha)\%$, that is the $1-\alpha/2$-quantile of a standard normal distribution. In our experiments, $l = 0.025$, $\alpha = 0.05$ and $z_{0.975} = 1.96$. The samples are drawn using the simple random methodology.

2.5.2 Results

Table 2.9 shows that both procedures are able to produce a synthetic population having a number of agents (both households and individuals) close to the real one. However, we observe that the new generator's figures are closer to the correct values.

	True	**New Generator**		**Guo & Bhat**	
		Generated	*APD*	Generated	*APD*
Individuals	10,635,691	10,634,902	< 0.001	9,731,686	0.085
Households	4,333,448	4,420,209	0.020	4,126,054	0.048

Table 2.9 – Generated agents by generator.

We pursue the comparison by reporting, in Table 2.10, statistics on the AAPD between the true and generated populations.

	Hh		**Ind**	
	New generator	Guo & Bhat	New gen	Guo & Bhat
Min	0.283	0.441	0.130	0.402
Max	6.807	13.188	0.603	2.511
Mean	0.596	0.665	0.322	0.658
Std dev	0.417	0.614	0.079	0.155
Median	0.533	0.578	0.314	0.630

Table 2.10 – AAPD statistics.

These results clearly favor the new generator, an observation confirmed by a one-way ANOVA analysis (with p-value smaller than 0.001 at a level $\alpha = 0.05$),

whose notched box-plots are shown in Figures 2.10 and 2.11. We also report in Figures 2.12 and 2.13 the maximum value of the APD per agent type. Again the conclusion favors the new approach, especially for the individuals.

At a more disaggregate level, Figures 2.14 and 2.15 give the same maxima but now computed for each agent type (sorted by class size) on the worst municipality in term of APD (separately for households and individuals) for each generator. Note that these figures are plotted using a logarithmic scale. As expected, the larger errors correspond to the smaller agent classes.

The goodness-of-fit of the distributions produced by both households' generation procedures with respect to the estimated ones is considered in Table 2.11. This table gives the proportion of municipalities for which the generated distribution of the agents' attributes is statistically similar to the estimated one at a 95% levels of confidence. The Freeman-Tukey statistic has been used to test the similarity. As one can see, both generator produce individuals' attributes joint-distributions for each municipality fitting accurately the estimed ones. This observation unfortunately no longer holds for the households' attributes joint-distributions. Indeed, while the distributions generated by the new generator still match the estimated ones, the IPFP-based approach performs poorly: less than 25% of the generated distributions adequately fit the estimated ones.

	Hh	**Ind**
New generator	100.0%	100.0%
Guo & Bhat	23.8%	100.0%

Table 2.11 – Proportions of municipalities statistically similar to the estimation ($\alpha = 0.05$).

Considering the agent's disaggregation level, Figures 2.16 and 2.17 illustrate the distribution of APDs' means and standard deviations for each individual and household type by generator. These provide some evidence that the new generator outperforms that of Guo and Bhat in terms of APD between the estimated and the generated agents' attributes joint-distributions.

Note that the true population to be reconstructed was formerly generated by the new generator, which may introduce a bias in its favour. Consequently, we conducted the same validation experiments with a true population generated by Guo and Bhat's methodology. These experiments results are highly similar to the previous ones and further illustrate the performances of the new generator. For instance, the proportion of municipalities for which the generated distribution of the agents' attributes is statistically similar to the estimation indicated in Table 2.12 (according to the Freeman-Tuckey goodness-of-fit test) does not differ significantly from the ones previously presented in Table 2.11.

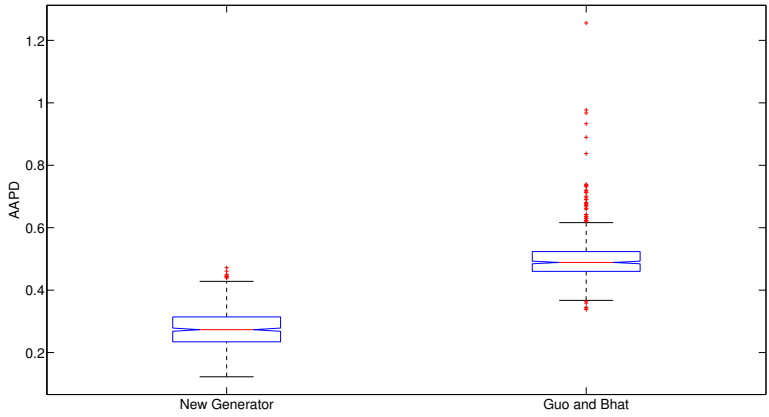

Figure 2.10 – Notched box-plot of the $AAPD$ for the individual's types.

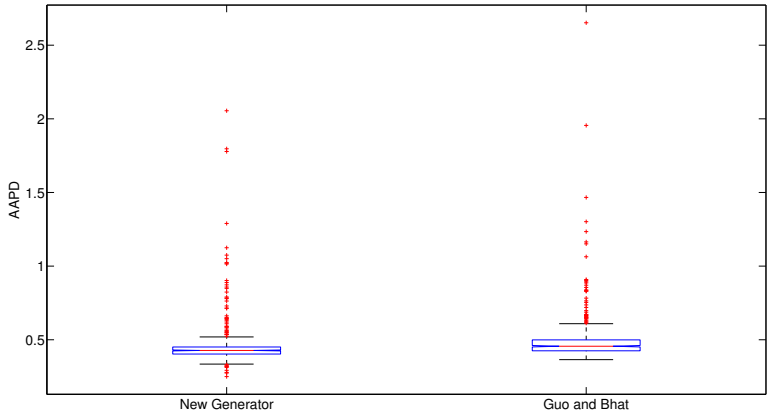

Figure 2.11 – Notched box-plot of the $AAPD$ for the household's types.

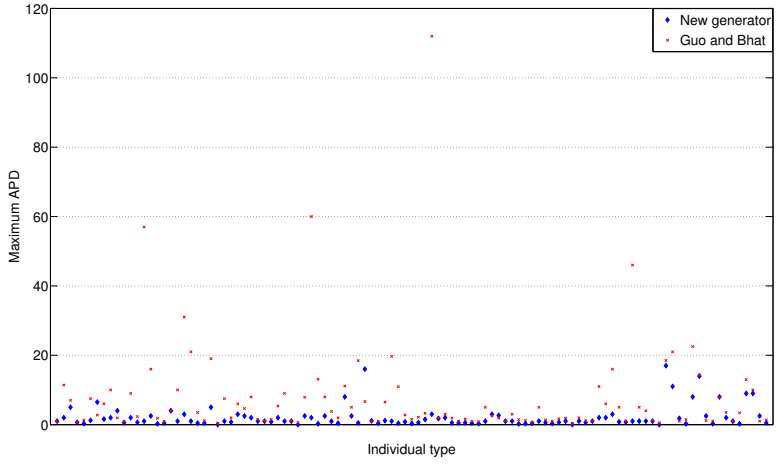

Figure 2.12 – Maximum APDs' repartition for disaggregated individual types.

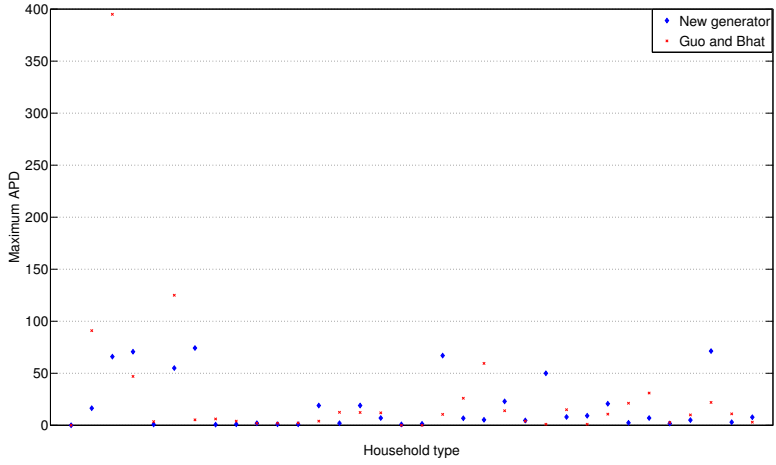

Figure 2.13 – Maximum APDs' repartition for disaggretated household types.

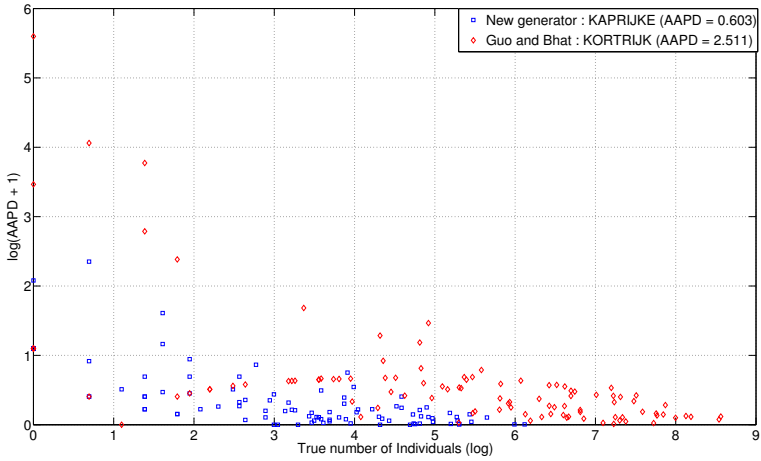

Figure 2.14 – AAPD vs. generated individuals by generator for the worst municipality (log scale).

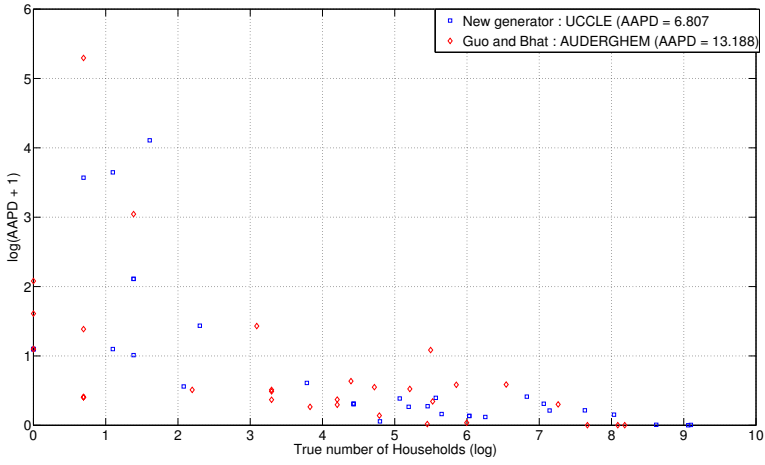

Figure 2.15 – AAPD vs. generated households by generator for the worst municipality (log scale).

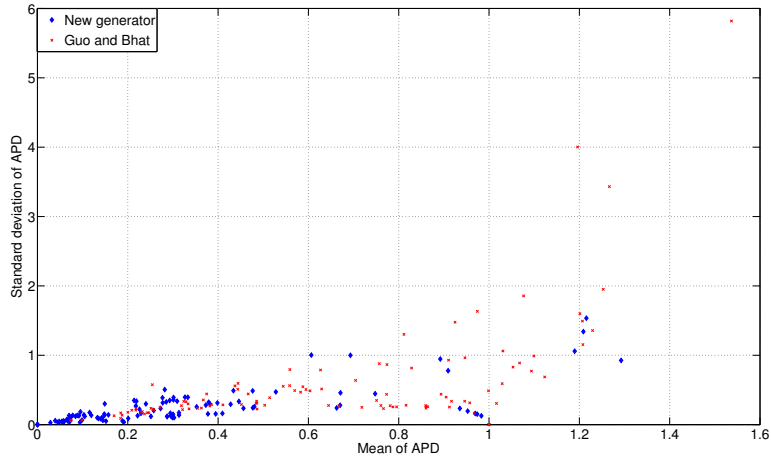

Figure 2.16 – $AAPD$s' means and standard deviations for each individual type.

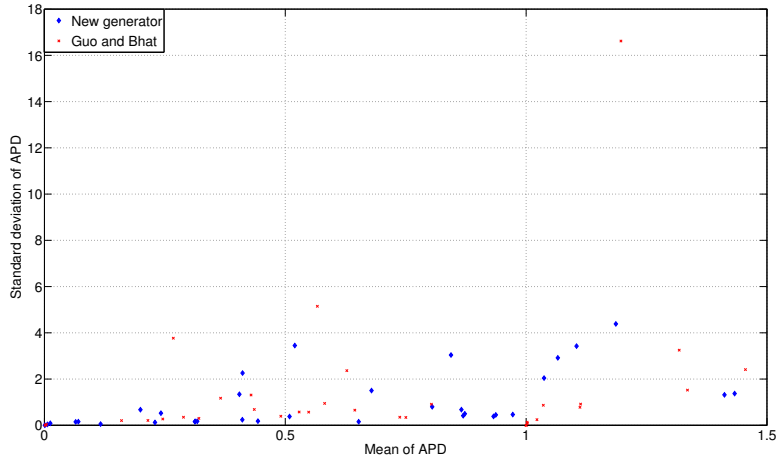

Figure 2.17 – $AAPD$s' means and standard deviations for each household type.

	Hh	Ind
New generator	100.0%	100.0%
Guo & Bhat	33.8%	100.0%

Table 2.12 – Proportions of municipalities statistically similar to the estimation ($\alpha = 0.05$, initial population generated by the method of Guo and Bhat).

2.5.3 Sensitivity analysis

The sensitivity of the proposed method with respect to data inconsistencies is also investigated in order to further assess the performance of the new generator. Different level of noise have been added to the data used for generating a synthetic population for Antwerpen (the largest Belgian municipality): the margins for all distributions have been corrupted by uniform relative random noise at 5%, 10% and 15% levels. Given the level of inconsistency reported in Section 2.2.3 for the population of interest, generating higher levels of inconsistency seemed excessive. The results are reported in Table 2.13 where the difference in AAPD values compared to values quoted in Section 2.5.2 is explained by the more disaggregate level considered. We note that the generator produces similar AAPD at the individual and household level when data inconsistencies remain of the order of 10%, which is similar to the level observed for the Belgian data. When noise increases, the individual's AAPD seem more stable than the household's ones.

Noise level	**AAPD**	
	Individual	Household
0%	0.193	0.402
5%	0.235	0.371
10%	0.226	0.305
15%	0.246	0.906

Table 2.13 – AAPDs' evolution for Antwerpen as a function of the noise level.

2.6 Conclusions

We have described in this chapter a new synthetic population generation technique, belonging to the class of SR methods, which is designed to overcome some limitations of IPFP-based methods. In particular, the generator is sample-free and can handle (moderate) data inconsistency which is common when data is extracted from several sources. Furthermore, its sample-free nature implies that it does not require an expensive survey to obtain the data needed for the generation or data protected by stringent privacy laws with may apply in some countries such as Belgium.

Micro-simulations using synthetic data as input are obviously influenced by the quality of the population generated, since the correlation structure of resulting synthetic population reflects that given by the available data sets, but not necessarily the true one. While it remains crystal-clear that every effort should be made to invest in quality data, this is not always possible (with the desired standards) and the question then arises whether one should simply give up analysis or try to accommodate the available data sources. If the second path is chosen, which is our option in this chapter, proportional care should be taken in interpreting the results.

The generator has been used to produce a synthetic population for Belgium at the municipality level. The results of the validation tests conducted on the households' generation procedure and the comparison with a more conventional approach indicate that the methodology has real potential to produce reliable synthetic populations. Consequently this synthetic population for Belgium will be the baseline for generating VirtualBelgium's agents used in the next chapter.

Chapter 3

A stochastic and flexible activity-based simulation

Contents

3.1 Introduction

Assuming that a synthetic population of individuals is available, we can now turn to the simulation of their traffic demand. Modelling this demand has been achieved with different methods but the current trend is grounding this modelling on activity patterns. These activity-based models form a class of travel demand forecasting models originally based on ideas by Hägerstrand (1970) and Chapin (1974). These were proposed as an alternative to the classical four-stages trip-based models for travel demand forecasting, whose successive steps are

- trip generation determining the frequency of trips origins and destinations (*i.e.* the marginal distributions of origin and destination);
- trip distribution matching origins with destinations;
- modal choice computing the transportation mode;
- and route assignment.

The drawbacks of these four-stage models were by then well identified (*i.e.* Dickey, 1983, Domencich and McFadden, 1975, Spear, 1977, Oppenheim, 1995). Their main criticism is they lack a valid representation of the underlying travel behaviour. Indeed they intrinsically ignore the travel demand as being derived from activity participation decisions and focus only on individual trips, ignoring the relations between all trips and activities.

On the other hand, activity-based approaches rely on the paradigm that people travel to carry out activities they need or wish to perform. Such models reflect the scheduling of activities performed by individuals in time and space and the sequence of activities, also names activity chains or activity patterns, becomes the relevant unit of analysis. This approach is now widely accepted and continues to attract a lot of attention.

Activity-based models can be classified in at least four families. The first two are discrete choice models (Adler and Ben-Akiva, 1979, Bhat and Koppelman, 1999, Bradley et al., 2010 and Bhat et al., 2004) and mathematical programming techniques (Gan and Recker, 2008). They have the drawbacks that the former approach may requires an extremely large choice set in order to capture a sufficient fraction of feasible mobility patterns, while the latter may not be tractable as the decision processes formulation may be extremely complex. This last issue also appears in structural equation modelling techniques, another family of activity-based models, which is rather confirmatory than explanatory. We refer the reader to Golob (2003) for a review of contributions using this approach and to Hoe (2008) for an insight on its limitations. Finally, the fourth model family exploits the advent of high performance computing: it uses massive multi-agents micro-simulations in order to reproduce behaviours within a complex system, such as mobility behaviours of a large population (Kitamura et al., 1997).

It has been noted that *"micro-simulation ... is drawing attention as a new approach to travel demand forecasting"* (Miller 1996), and several operational micro-simulators for activity scheduling are currently in use. Examples include ALBATROSS (Arentze and Timmermans, 2000) for the Netherlands, TASHA (a part of the ILUTE simulator, Salvini and Miller, 2005) for the Greater Toronto Area, SAMS and AMOS (Kitamura et al., 1996). A review and comparison of various micro-simulators and discrete choice models for activity-based modelling can be found in (Goran, 2001) and (Benromach et al., 2014). These approaches typically implement the first three steps (generation, distribution and modal choice) of the traditional four-stage model. The last step, namely traffic assignment, can be handled with dynamic traffic assignment procedures, whose adoption has been made easier by the the development of powerful open source agent-based simulation systems such as MATSim, used by Meister et al. (2010) in travel demand forecasting for Switzerland, Urbansim (Waddell, 2002) and Transims (Nagel et al., 1999).

Even though all these approaches have demonstrated their usefulness, they typically require, in addition to a complete description of the road network, an a-priori localisation of every housing unit, service, shop,... This turns out to be a strong requirement: indeed, if this information can often be gathered for a particular city or even a district of a country, the geo-localisation process is far more complex and cumbersome for a whole country and may not be feasible. This issue motivates our interest for the design of an alternative methodology obviating this limitation, but flexible enough to use every information available and making it suitable for a nationwide application.

The approach taken here is the micro-simulation of the Belgian population mobility behaviours as a part of the VirtualBelgium integrated simulator. The proposed activity scheduling model is a three steps procedure: first, a set of feasible activity chains is generated for every agent type; a chain is then assigned to every individual agent in the simulation using a randomized model; and all activities characteristics of the chain are finally determined based on statistical distributions. The outputs of the model can then be processed using MATSim for dynamic traffic assignment, if required. VirtualBelgium activity-based model mainly relies on data extracted from the Mobel and Beldam national mobility surveys conducted in Belgium (Hubert and Toint, 2002 and Cornelis et al., 2012) and the OpenStreetMap project (Haklay and Weber, 2008).

The remainder of this Chapter is organized as follows. Section 3.2 introduces the concept of activity chains possibly performed by individual agents, the underlying assumptions and data source used by this part of VirtualBelgium. In Section 3.3, we detail the proposed method for assigning activity chains to individual agents. We next present in Section 3.4 the results obtained with this methodology when applied to VirtualBelgium. Concluding remarks and future perspectives are finally discussed in Section 3.5.

3.2 Activity chains, general assumptions and data source

Activity chains data used by VirtualBelgium is derived from the Mobel 2001 and Beldam 2012 mobility surveys conducted in Belgium. These surveys highlighted 12 base activities:

d	pick up/drop	e	school	f	visiting relatives
m	staying home	r	eating outside	b	going for a walk
v	work related visit	c	shopping	l	leisure activity
t	work	p	personal reason	o	other

Each activity is also characterized by a duration and a localization within the Belgium's road network. Such a network is extracted from OpenStreetMap and is defined by the pair $G = (N, L)$, where N and L correspond respectively to the sets of nodes and links which can be tough respectively as crossroads or junctions and (portion of) roads. An activity localization is then a node of the network.

Note that individuals below 5 years old (included) are discarded as it is assumed that they always travel with their relatives and they do not have proper activity chains.

An activity chain is then a sequence of the base activities. It is assumed that each activity chain begins and ends at the individual's home. These concepts are formally described in Definition 1.

Definition 1 (Activity chain) *An activity α performed by an individual is a quadruplet $(\alpha^p, \alpha^l, \alpha^s, \alpha^d)$ where*

- α^p = *the purpose;*
- α^l = *the localization;*
- α^s = *the starting time;*
- α^d = *the duration;*

of the activity. An activity chain $\alpha^* = (\alpha_n)_{n\in\{1,\dots,k\}}$ of size k is then a sequence $\alpha_1, \cdots, \alpha_k$ of activities such that $\alpha_1^l = \alpha_k^l$.

The variety of observed activity chains is significant, as approximately 10,000 different such chains have been extracted from the mentioned national surveys.

3.3 Activity chains generation and assignment

How to assign activity chains to each individual in the VirtualBelgium simulation? This section presents a proposal for performing this assignment which does not rely on the geo-localization of each of the potential activity sites, an information which is (unfortunately) missing in our context. We start by outlining the main steps of our approach before a more formal description.

The first step is to generate a set of feasible activity-chains for each individual type available. It is also required that every individual is assigned to a house localized in the network, a task which is necessary because the synthetic population generator only specifies the homes' municipality. This house will be the starting and ending point of the activity chain for each individual living inside it. Once these preliminary steps have been performed, the assignment of a fully characterized activity chain to an individual consists in drawing an activity chain α^* from the appropriate activity-chains set and finally determining the characteristics of every activity $\alpha \in \alpha^*$. This methodology is fully described in the remainder of this Section.

3.3.1 Generation of activity chains patterns by individual type

In our context, an individual is characterized by a vector of m attributes $V = (V_1, \ldots, V_m)$, whose components may take a discrete and ordered set of values (see Table 2.3 in Section 2.4.1). Let us denote by $\mathcal{T}_\mathcal{I}$, $\mathcal{A}_i$ and n_i the set of all individual types, the set of activity chain patterns that could be extracted from the data relative to $i \in \mathcal{T}_\mathcal{I}$ and the size of $\mathcal{A}_i$, respectively. Definition 2 introduces the concept of neighbourhood for an individual type by shifting its attributes values.

Definition 2 (l-neighbourhood) *For an individual type i and a integer $l \in \{1, \ldots, m\}$, the l-neighbourhood of i, denoted by $\mathcal{N}_i^l$, is the set of all individual types obtained by at most l shifts between contiguous values of the attributes of type i.*

Depending on the data, the number of observed activity chains may be lower than a desired minimal threshold t for a subset of individual type $\mathcal{T}_\mathcal{J} \subseteq \mathcal{T}_\mathcal{I}$. It is then necessary to add activity chains to the problematic $\mathcal{A}_j$ such that the constraint

$$n_j \geq t \qquad \forall j \in \mathcal{T}_\mathcal{J} \tag{3.1}$$

yields. We propose to augment $\mathcal{A}_j$ with the activity chains in $\mathcal{A}_k$, where $k \in \mathcal{N}_j^l$ and l is as small as possible.

For VirtualBelgium, a threshold value t arbitrarily set to 5 has shown to produce reasonable diverse results. As one can observe in Figure 3.1, out of 192 individual types, 116 problematic classes were identified in the raw data and an at most a 3-neighbourhood was required to satisfy the constraint. In our

implementation, the $\mathcal{N}_j^l$ $(l = 1, 2, 3)$ are generated by sequentially modifying the following attributes:

1. gender;
2. gender and age class;
3. gender, age class and education level.

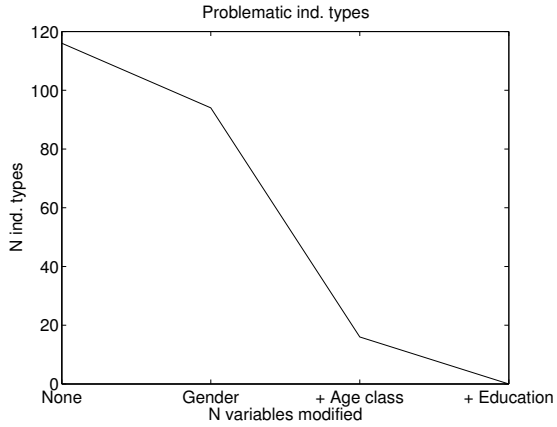

Figure 3.1 – Number of problematic individual classes with respect to the neighbourhood's level.

3.3.2 Activity chains assignment

Once a set of activity chains $\mathcal{A}_i$ is available for each individual type $i \in \mathcal{T}_\mathcal{I}$, the next step is to assign a chain to every individual agent. This is done by randomly drawing an activity chain α^* in $\mathcal{A}_i$ if the considered individual is of type i, using the empirical distribution obtained from the Mobel survey. For instance, Table 3.1 illustrates the set $\mathcal{A}$ of feasible activity chains and their respective weights for a student woman between 18 and 39 years old with a higher education degree and without a driving licence.

Pattern	m e b m	m f m	m e m	m e m b m	m e r e m	m l m
Weight	0.272	1.025	0.913	0.412	0.412	0.284

Table 3.1 – Weighted $\mathcal{A}$ for a given individual agent (Mobel).

3.3.3 Household's house localization

As stated previously each household and its constituent members are already located in one of the 589 municipalities. Nevertheless, as the goal is to locate an activity at network-node level, and since no data is available at a more disaggregate level, the first part of the process consists in assigning each household to a node of its municipality road network $G_{mun} = (N_{mun}, L_{mun})$, where $N_{mun} \subseteq N$ and $L_{mun} \subseteq L$.

The node, randomly drawn in N_{mun} following a discrete uniform distribution, meaning that every node belonging to N_{mun} has the same probability to be chosen (in order to preserve the population density of the municipality), will be referred as the household house.

3.3.4 House departure time

The first step taken by an agent is to leave its home in order to perform the first activity of the day, that means that a house departure time h must be determined. Regarding the activity type to be performed, the time departure distribution H varies and is approximated by a mixture distribution which is fitted to the empirical distribution obtained from the Mobel survey. The mixture is of the form

$$H \sim f(x \mid p) = \sum_{i=1}^{l} w_i C_i(x; \mu_i, \sigma_i^2 \mid p)$$

where p is the activity purpose, l the number of components, w_i is the weight associated with the component C_i such that $w_i \geq 0$ and $\sum_i w_i = 1$. Every C_i considered here follows a Log-Normal distribution $\mathcal{LN}(\mu_i, \sigma_i^2)$ with location parameter $\mu_i \in \mathbb{R}$ and scale parameter $\sigma_i^2 \in \mathbb{R}$. For a detailed description of such mixture distributions, see McLachlan and Peel (2000).

The empirical and fitted distributions are illustrated in Figures 3.2 and 3.3. It is important to note that the number of components is determined such that each mixture distribution obtained is statistically similar to the empirical distribution according to the univariate Kolmogorov-Smirnov goodness-of-fit test (Massey, 1951) at 5% significance level.

The departure time is then randomly drawn accordingly to the appropriate distribution.

Note that directly drawing from empirical distribution has also been investigated, but this was up to 3 times slower in the conducted experiments. As a very large number of draws is involved (in the hundred of millions), random number generation speed becomes an essential issue to address and this approach has been discarded.

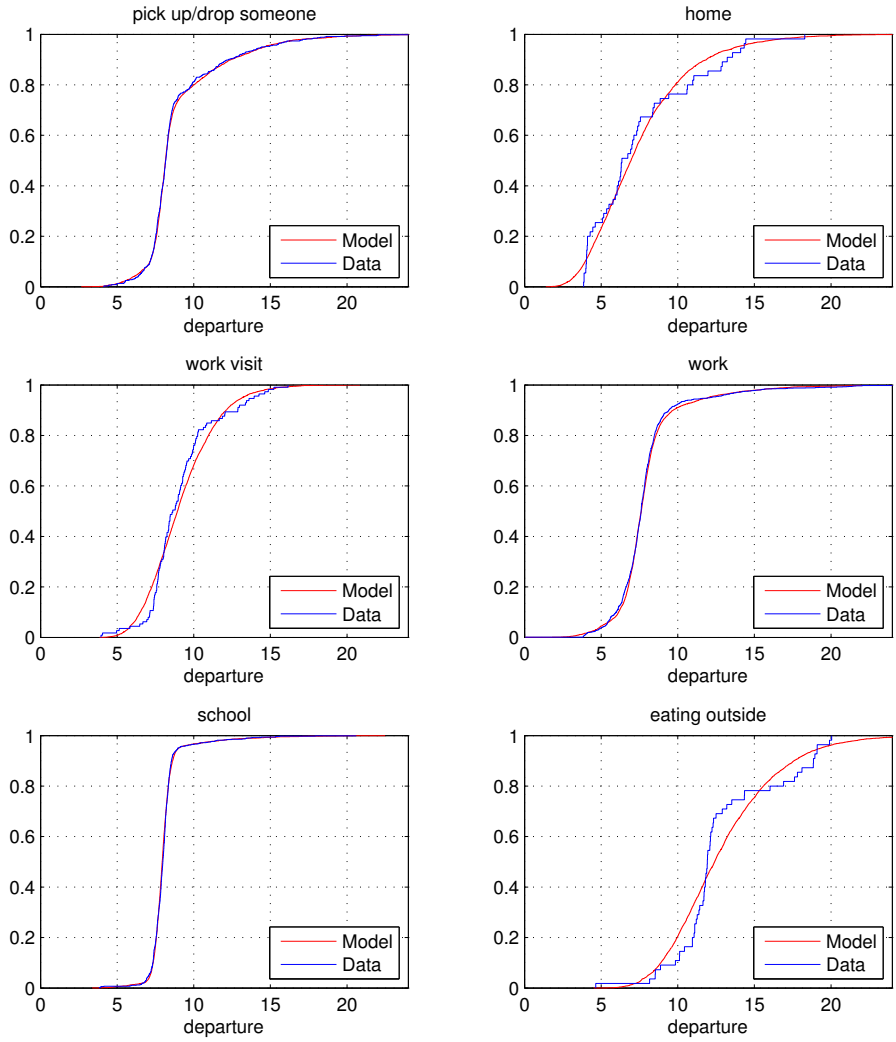

Figure 3.2 – House departure time (hours) : empirical and estimated cumulative distribution functions by purpose.

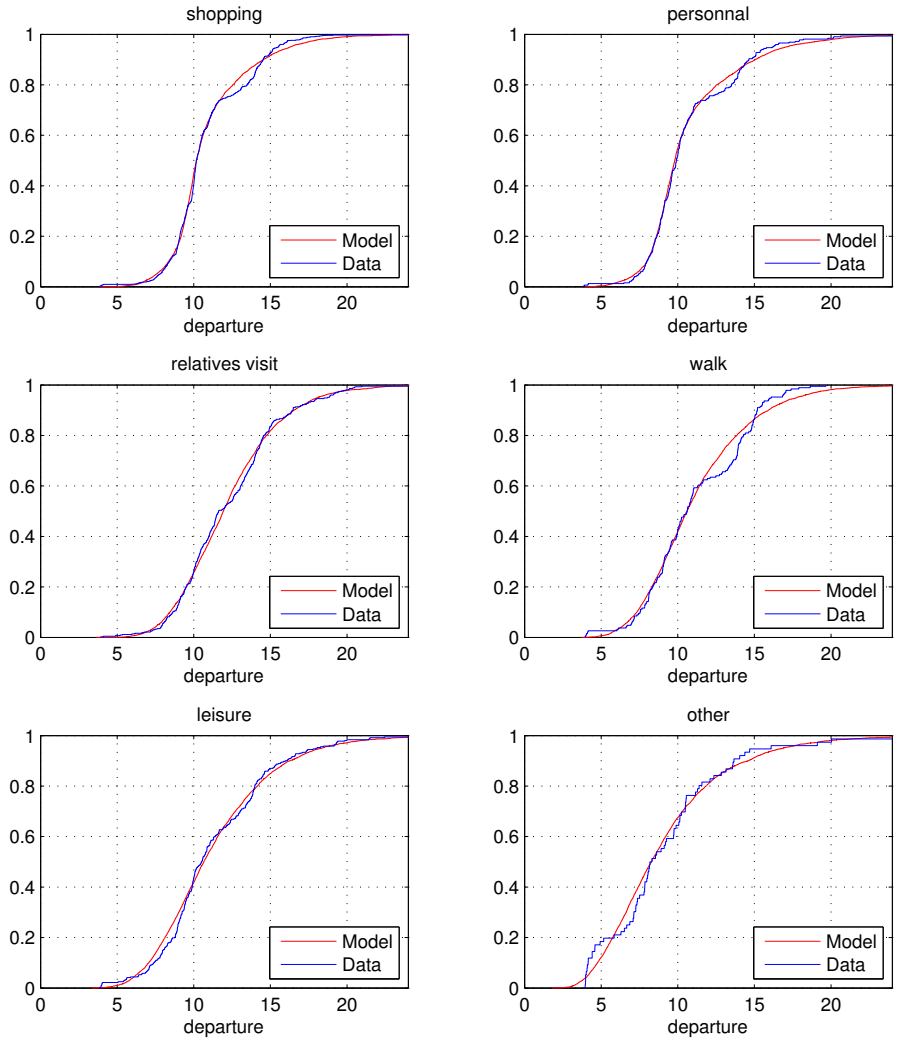

Figure 3.3 – House departure time (hours) : empirical and estimated cumulative distribution functions by purpose.

3.3.5 Activity localization

We now turn to the description of how the localization of an activity is determined inside the road network of Belgium. Given that each individual has a house, it is possible to localize each of his/her activities in the network, the house being the starting point of the activity chain. These activities will also take place at a node of the network, which is determined as follows:

1. a distance d is drawn from a distribution pertaining to the considered activity;
2. a set of nodes at distance d from the current localization is generated;
3. finally a node is drawn from the set generated at previous step.

We now give more details on these three steps.

3.3.5.1 Random draw of a distance

Similarly to the house departure time, the random draw of the distance d travelled to perform an activity follows a mixture of distributions conditional to the type of the activity which is fitted to the empirical distribution obtained from the Mobel survey. Empirical and resulting fitted probability density functions are illustrated in Figures 3.4 and 3.5.

3.3.5.2 Generation of a set of feasible nodes for a given activity

The next step is to generate a set of nodes at distance d from the current localization, at which the considered activity could take place. A Dijkstra algorithm relying on a *Fibonacci heap* data structure, is used to explore the network and find these feasible nodes. For a network with n nodes and m arcs, this algorithmic variant has the crucial advantage in our context of requiring $\mathcal{O}(n \log n + m)$ operations, instead of $\mathcal{O}(n^2)$ for a more direct implementation or $\mathcal{O}((n+m)\log n)$ for an implementation based on a k-ary heap data structure[(1)] (Fredman and Tarjan, 1987). If no suitable node is found at the desired distance, then the same procedure is applied but with a range of distances $[d-\epsilon, d+\epsilon]$. This error term ϵ is increased[(2)] until at least one node is discovered.

3.3.5.3 Activity node choice

If no additional data is available, the destination node α^l is then randomly chosen from a discrete uniform draw. Otherwise, the draws can be empirically weighted in order to take information on specific activity localization at specific nodes/municipalities (for instance using geo- localization) into account. To

[(1)] this computational advantage was confirmed in the conducted experiments as the Fibonacci heap data structure was up to 50× faster than the others ones

[(2)] in practice, doubled with initial value of 250m

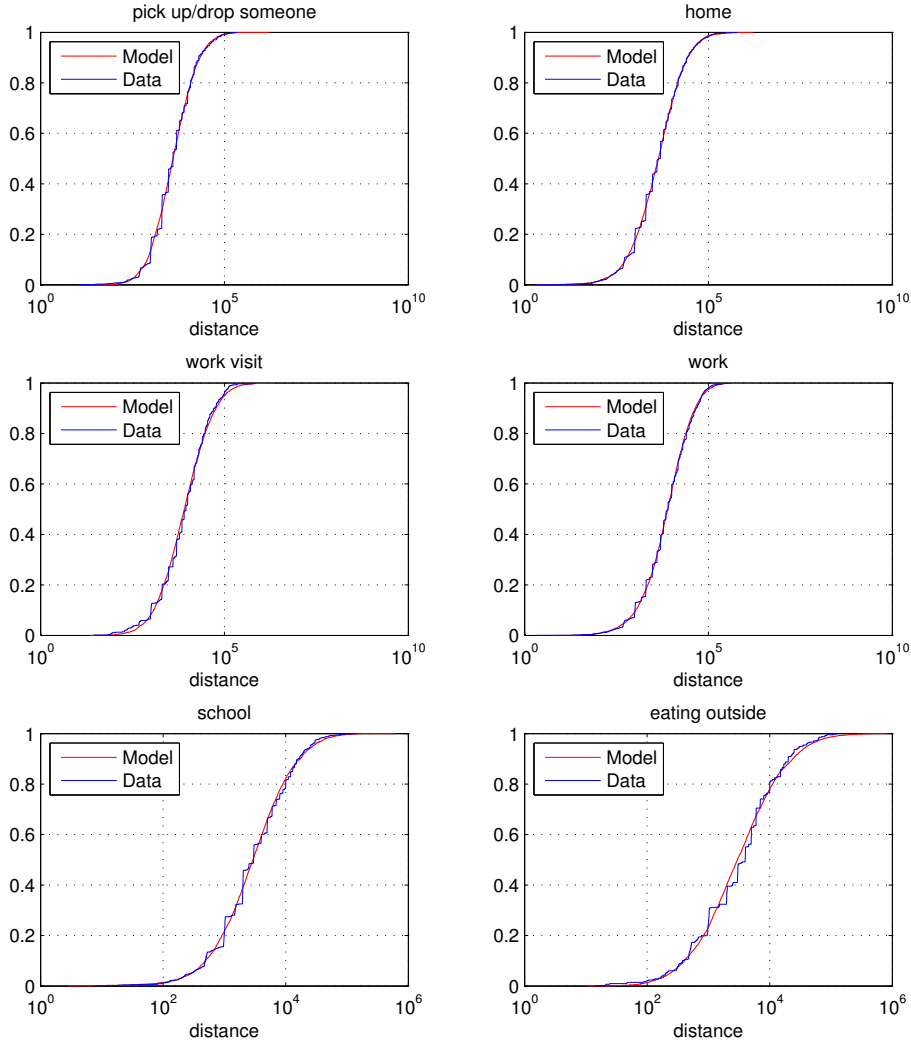

Figure 3.4 – Distance (meters) : empirical and estimated cumulative distribution functions by activity type.

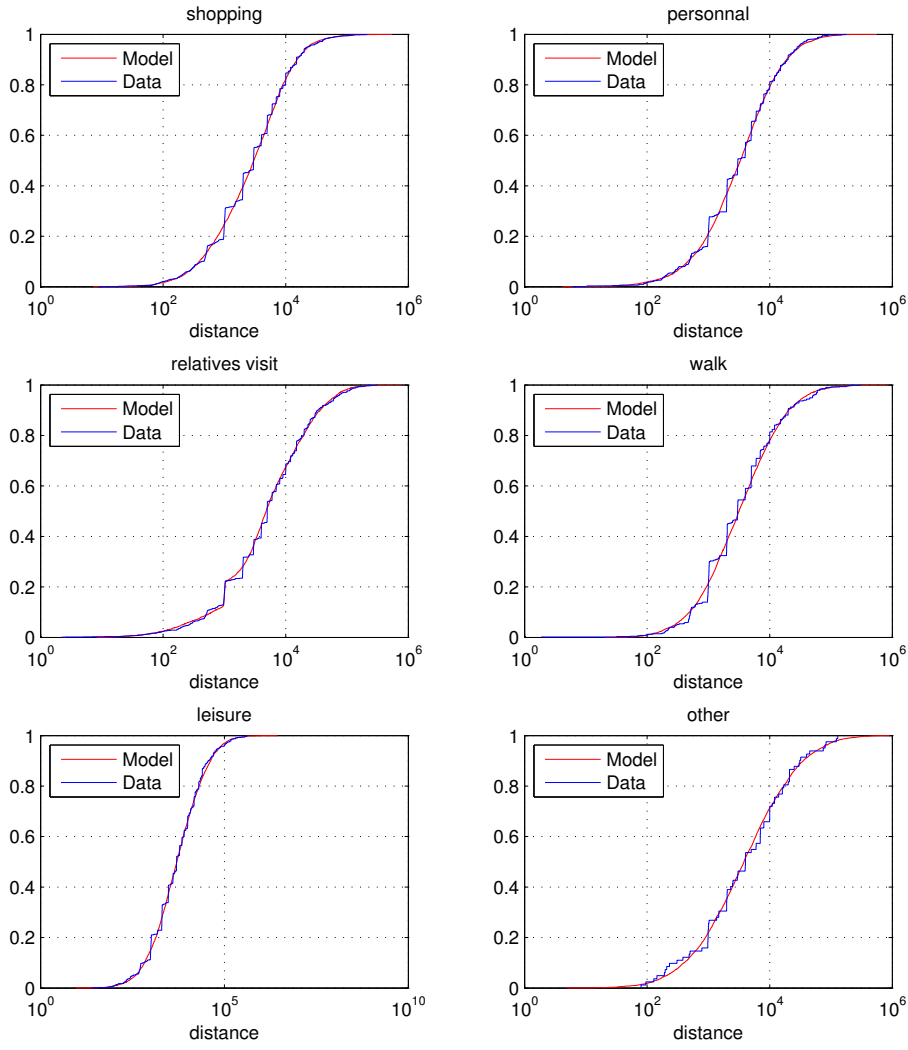

Figure 3.5 – Distance (meters) : empirical and estimated cumulative distribution functions by activity type.

illustrate our proposal, assume a road network and an activity choice resulting in a set of 4 feasibles nodes, whose indicators for 3 types of activity are detailed in Table 3.2 (nodes 1 and 2 belongs to the same municipality). If no indicator is available, such as for *leisure*, then the line is set to *na*; *work* is a municipality-related indicator and *school* is a node-related indicator used for precise geo-localization of schools.

Indicator	Node 1	Node 2	Node 3	Node 4
work	1000	1000	500	800
school	0	1	0	0
leisure	na	na	na	na

Table 3.2 – Nodes' indicators.

The proposed technique has the advantage of using localization data whenever available, but also allows for a reasonable alternative, would such information be missing.

3.3.6 Activity duration

An activity duration depends on its starting time, which is obtained by adding the ending time of the previous activity and the trip duration performed to reach the current localization. The time spent to carry out an activity is then determined by

1. drawing a trip duration t to compute a starting time α^s;
2. and drawing an activity duration α^d conditional to s.

These two steps are detailed in the remaining of this Subsection.

3.3.6.1 Trip duration and starting time

It is clear that a trip duration t is related to its distance d. This observation, confirmed by the data extracted from Mobel, leads us to fit a mixture of bivariate distributions to approximate the joint-distribution of (D, T) where T and D are the random variables respectively associated with the duration and the distance of the trip. The resulting bivariate distribution is illustrated in Figure 3.6 and is defined by

$$(D, T) \sim f(\boldsymbol{x}) = \sum_{i=1}^{l} w_i C_i(\boldsymbol{x}; \boldsymbol{\mu}_i, \Sigma_i)$$

where l is the number of components, w_i is the weight associated with the component C_i such that $w_i \geq 0$ and $\sum_i w_i = 1$. The C_i considered here follow a bivariate Log-Normal distribution $\mathcal{LN}(\boldsymbol{\mu_i}, \Sigma_i)$ with location vector

$$\boldsymbol{\mu_i} = (\mu_{i,1}, \mu_{i,2})$$

and scale matrix

$$\Sigma_i = \begin{pmatrix} \sigma_{i,11} & \sigma_{i,12} \\ \sigma_{i,21} & \sigma_{i,22} \end{pmatrix}.$$

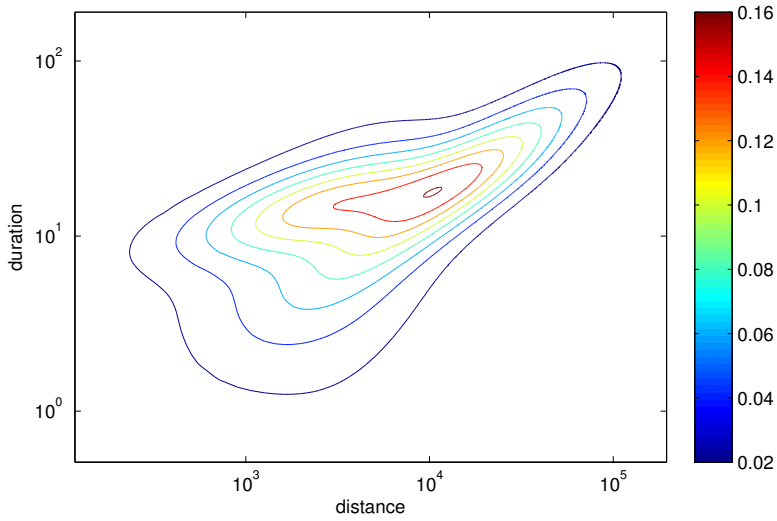

Figure 3.6 – Fitted probability density function of distance (meters) × duration (minutes).

As for the distributions of the house departure time and the distance performed to reach an activity, the number of components l is determined in order to obtain a fitted distribution that is statistically similar to the empirical distribution according to the Fasano and Franceschini's generalization of the Kolmogorov-Smirnov's goodness-of-fit test (Fasano and Franceschini, 1987) at significance level of 5%.

The fitted distribution is illustrated in Figure 3.6. As one could expect, there is a positive correlation between the distance and the duration of trip, *i.e.* the further an individual goes, the more time he spends on the road. It can also be noted that the variance of the duration is higher for smaller trips, and gradually decreases as the distance increases.

Since the distance d is computed in Section 3.3.5, it follows from (Eaton, 1983) that the trip duration t can be draws from the univariate conditional distribution of T given $D = d$ defined by

$$T \mid D = d \sim f(x \mid D = d) = \sum_{i=1}^{l} w_i C_i(x; \mu_i, \sigma_i)$$

where l is the number of components, w_i are the weights of the mixture and

C_i follows a univariate Log-Normal distribution $\mathcal{LN}(\mu_i, \sigma_i)$ such that

$$\mu_i = \mu_{i,2} + \frac{\sigma_{i,12}}{\sigma_{i,11}}(d - \mu_{i,1})$$

and

$$\sigma_i = \sigma_{i,22} - \frac{(\sigma_{i,12})^2}{\sigma_{i,11}}.$$

The starting time of $\alpha_i \in \alpha^*$ $(i > 1)$ is then obtained by adding the transportation duration and the ending time of the previous activity of the chain, *i.e.*

$$\alpha_i^s = (\alpha_{i-1}^s + \alpha_{i-1}^d) + t.$$

3.3.6.2 Activity duration

Since an activity duration is correlated with its starting time and purpose, the computation of α^d follow a similar process applied for determining a trip duration, *i.e.* for each purpose the joint- distribution of an activity starting time and its duration is fitted to the data. Figure 3.7 and 3.8 illustrate the resulting joint-distributions, from which behavioural patterns can be observed. For instance

- individuals mainly start working at 8:30 for 8 to 9 hours, but the distribution also highlights the part-time worker starting at 8:30 or 13:00;
- students usually start the school between 8:00 and 8:30, and remains there either 4 hours (on Wednesday) or 8 hours (the other school opening days). Also the later a student arrives at school, the less time he spend there;
- eating outside occurs at midday and in the evening. An average midday and evening lunch takes 1:20 hour and 2:15 respectively. This indicates that midday lunch duration is more constrained by the time budget available for the remaining activities of the day;
- shopping is mainly done before midday (between 10:00 and 11:00) and around 16:00 *i.e.* after leaving or work, with a typical duration of 30 minutes to 1 hour.

These observations indicate that the fitted distributions produces consistent behaviours.

A duration α^p is then draws from the distribution pertaining to the considered activity purpose conditionally to the starting time computed previously.

Finally, the activity chain of the individual is completed by generating a return to home after the end of the last activity.

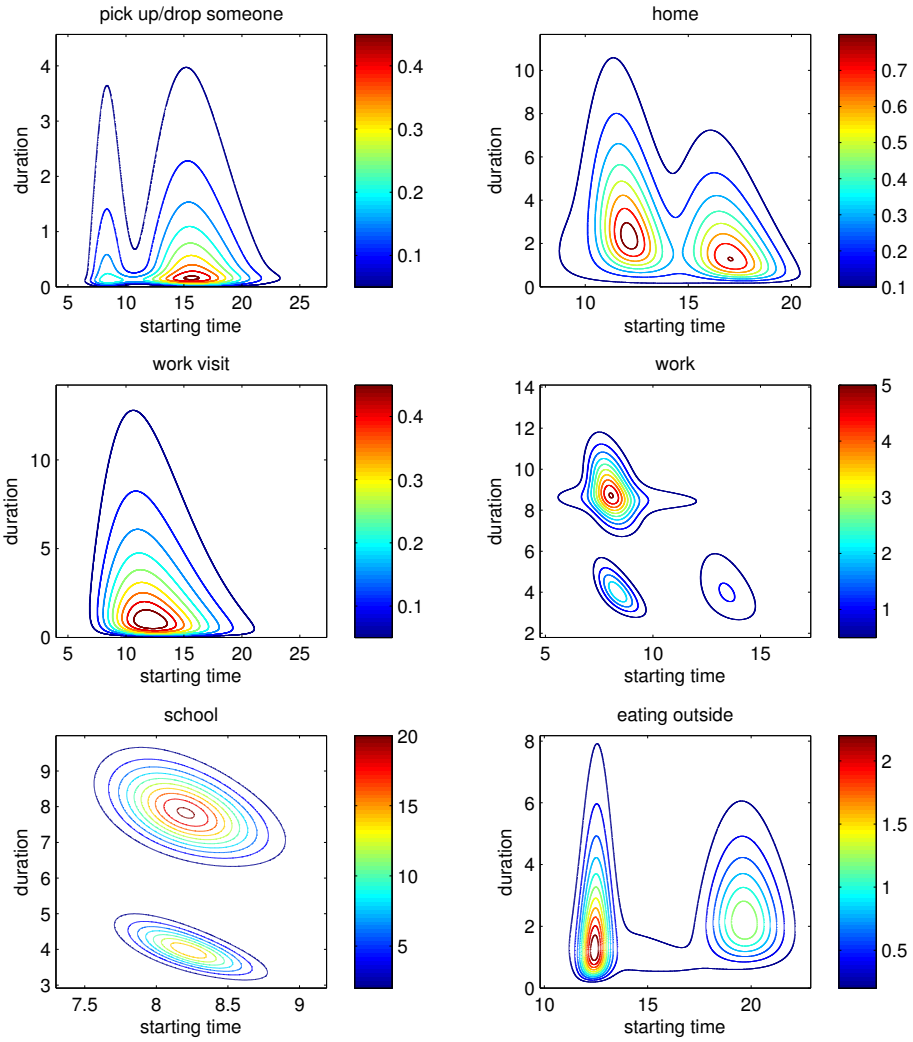

Figure 3.7 – Starting time × Duration (hours) : estimated probability density functions by purpose.

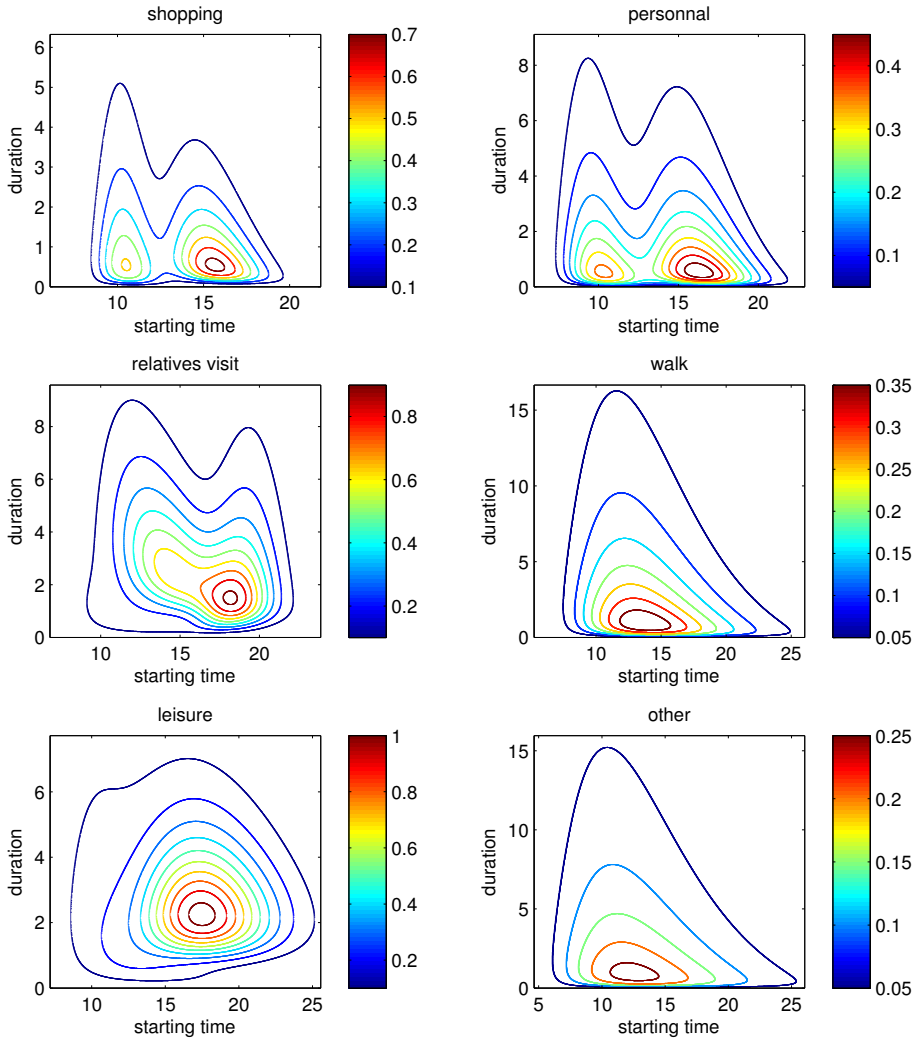

Figure 3.8 – Starting time × Duration (hours) : estimated probability density functions by purpose.

3.4 Application on VirtualBelgium: results

Our activity-based model has been successfully applied to the Belgian synthetic population to simulate an average day. As previously stated in Chapter 2, the simulation involved 10,300,000 agents spread in 4,350,000 households. With an average of 4.33 activities per individual, we have 43,300,000 activities to characterize. The road network considered is illustrated in Figure 3.9, which is made of 66,304 nodes and 125,889 links. It is detailed up to the OpenStreetMap tertiary road network.

Figure 3.9 – Belgian road network - 66.304 nodes and 125.889 links.

The size of the simulation generates a substantial amount of computation, whose efficient organization and structuration are truly challenging. The main computational burden is the execution of many shortest-path calculations for activity localization, as well as efficient random draws. Our current best execution time is approximatively 11:00 hours using a cluster of 500 Intel(R) Xeon(TM) X5650 processors' cores and 1GB of RAM per core[(3)].

[(3)] computational resources have been provided by the Consortium des Équipements de Calcul Intensif (CÉCI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under Grant No. 2.5020.11

Figure 3.10 shows the histogram of proportions of activities starting at each hour of the day. One can easily observe the morning and evening peaks occurring at 8:00 and 16:00. The comparison, respectively for the cumulative distribution function and the probability density function, between the Mobel data and VirtualBelgium are given in Figures 3.11 and 3.12. The Kolmogorov-Smirnov's test indicates that these distributions are not significantly different; the minor differences may result from correlation structures not taken into account implying that the agents can then accumulate delays over the day. This result is crucial since it shows that, at an aggregate level, the VirtualBelgium agents behave as expected.

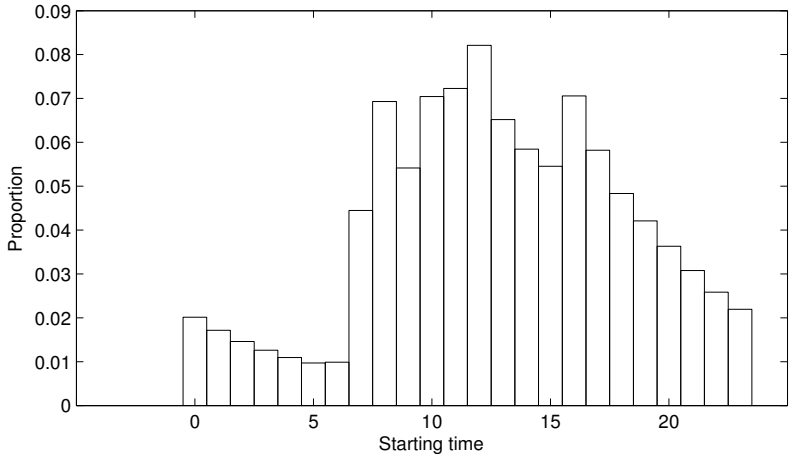

Figure 3.10 – Histogram of the number of the number of activity starting at each hour of the day.

The difference between VirtualBelgium and Mobel in proportion of activity is presented at Figure 3.13. One can easily see that the differences remain very small, with a maximum difference less than 10%. This observation seems to validate the generation of activity chain patterns by individual type and the assignment process.

An interesting output of the model is the origin-destination matrix that identifies the number of trips between municipalities. The total flows for one average day is represented in Figure 3.14. As expected, the main cities of Belgium attract most part of the traffic flows. This can also be observed in the maps in Figures 3.15 to 3.38 illustrating the number of activities starting each hour of an average day by municipality. This result is encouraging as no node indicator (as defined in Section 3.3.5.3) has been used. This is certainly explained by the fact that these cities have a more dense road network (and therefore have more nodes than smaller cities), thus the activity localization process naturally favours them.

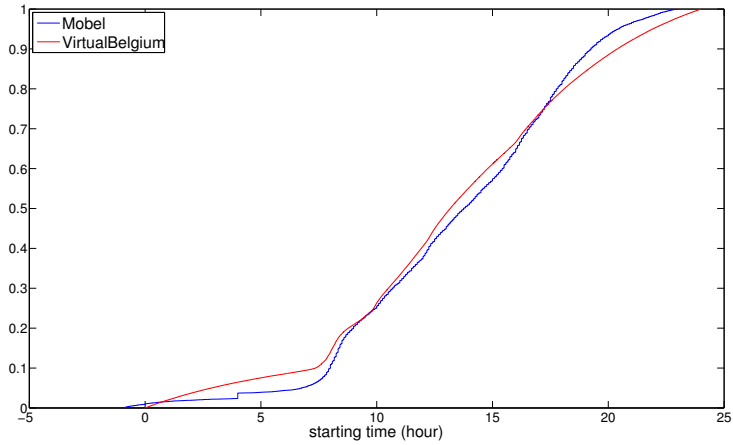

Figure 3.11 – Comparison of the empirical and resulting cumulative distributions.

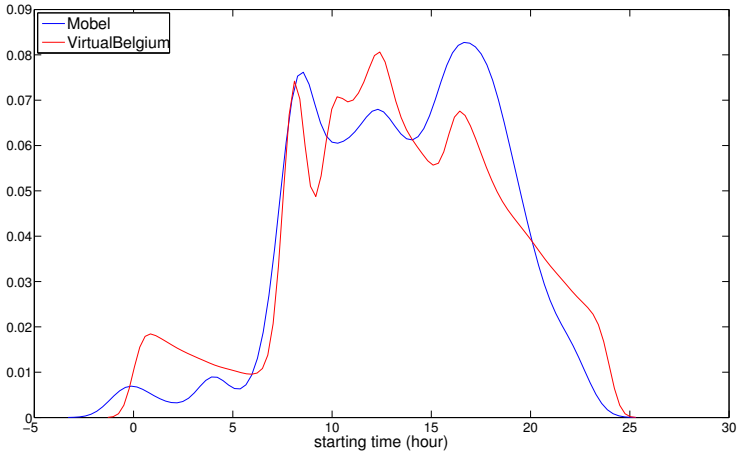

Figure 3.12 – Comparison of the empirical and resulting probability density distributions.

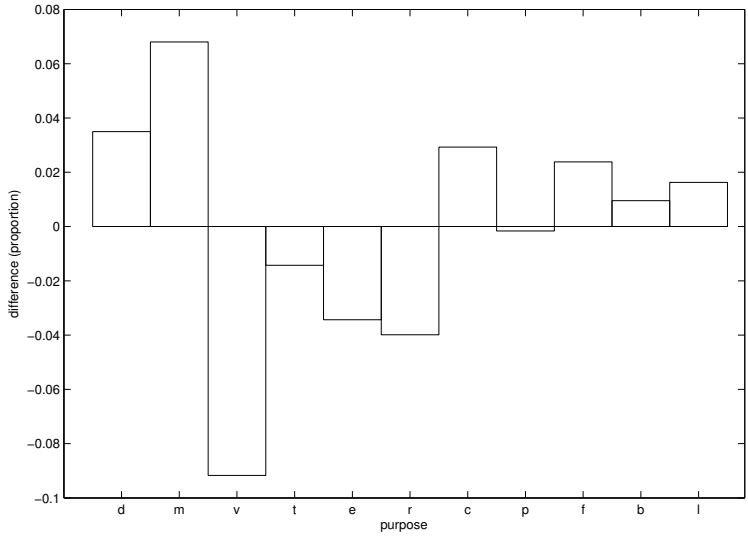

Figure 3.13 – Difference of activity type proportions between VirtualBelgium and Mobel.

Figure 3.14 – Origin-destination flows between municipalities.

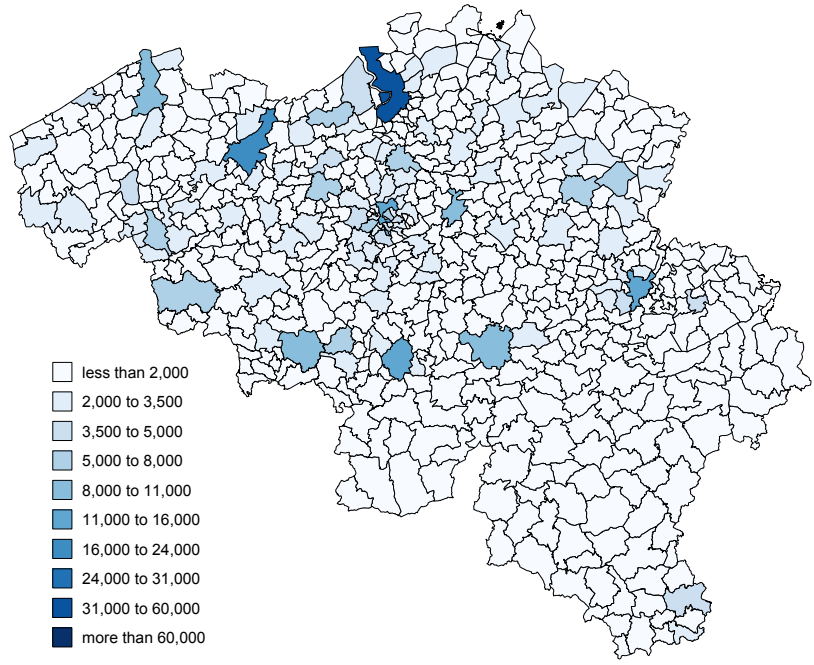

Figure 3.15 – Starting activities by municipality between 0:00 and 1:00.

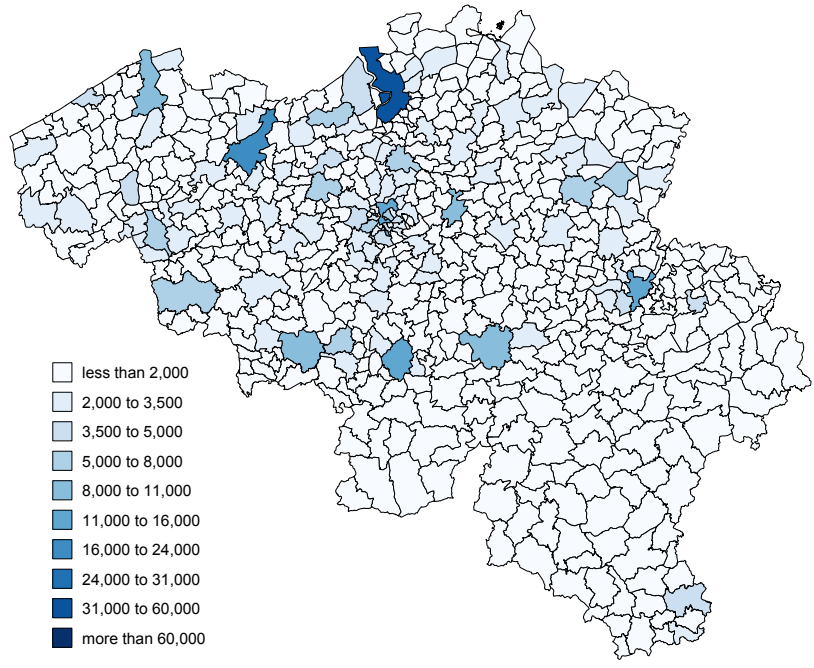

Figure 3.16 – Starting activities by municipality between 1:00 and 2:00.

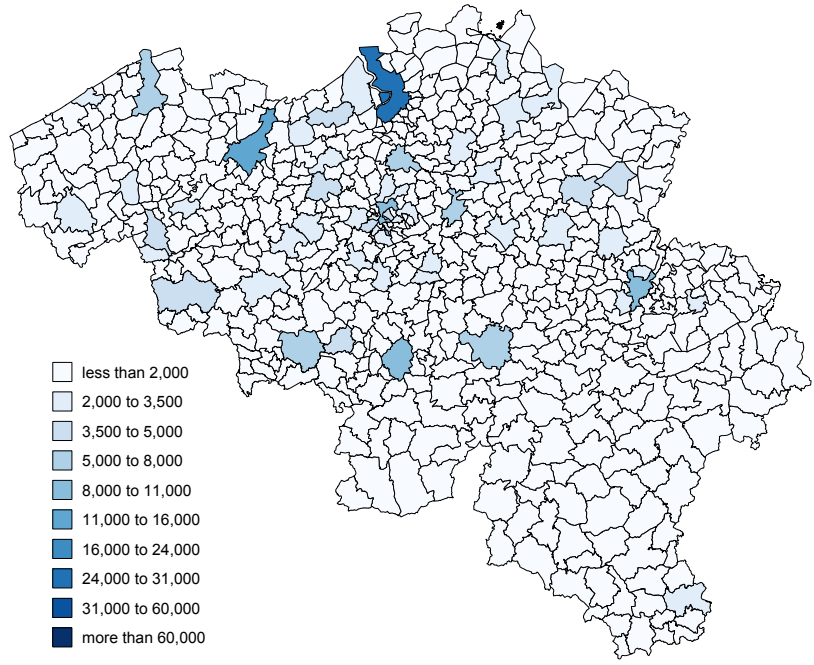

Figure 3.17 – Starting activities by municipality between 2:00 and 3:00.

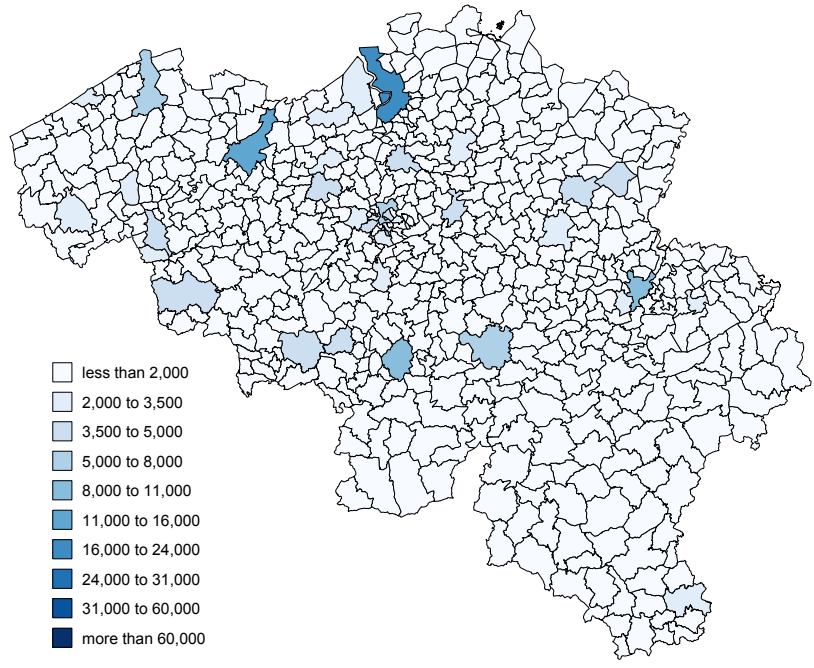

Figure 3.18 – Starting activities by municipality between 3:00 and 4:00.

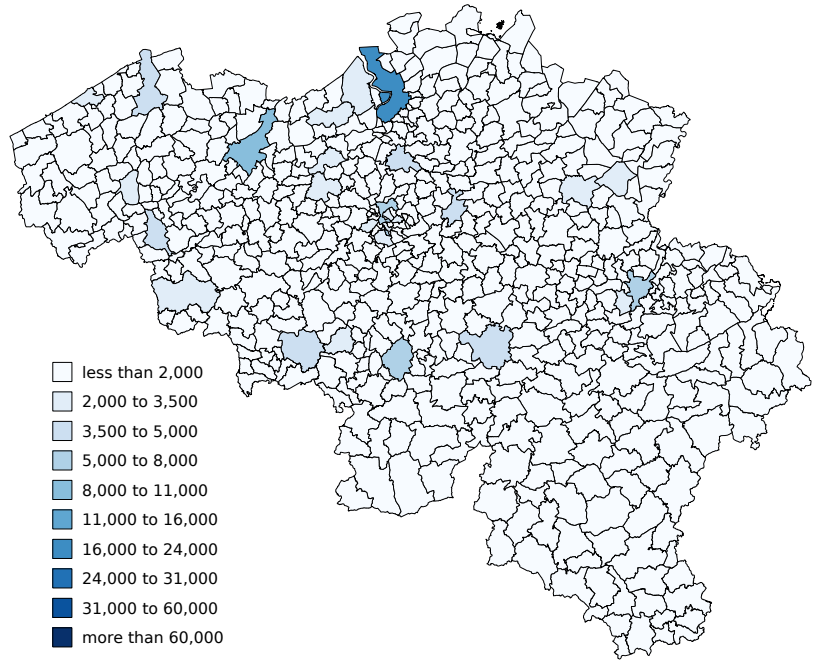

Figure 3.19 – Starting activities by municipality between 4:00 and 5:00.

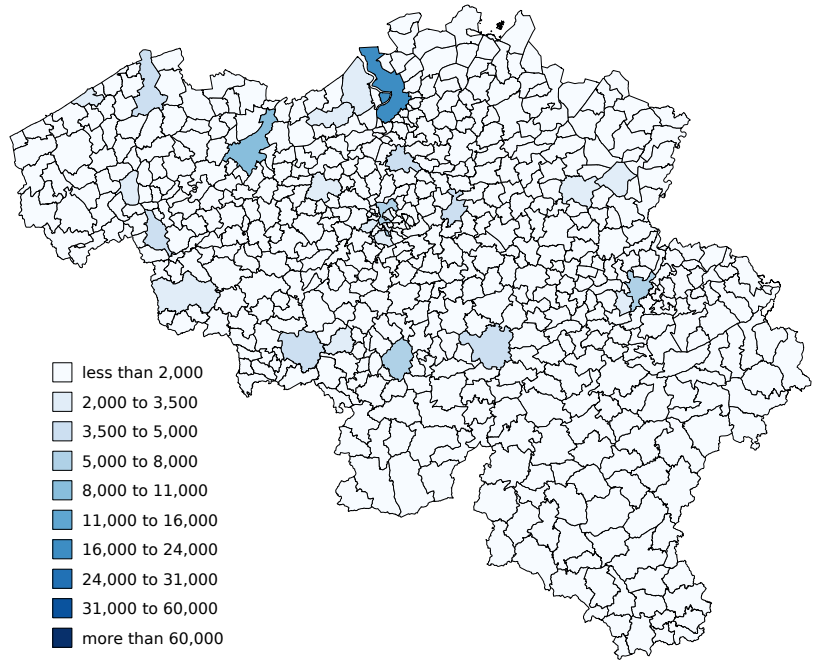

Figure 3.20 – Starting activities by municipality between 5:00 and 6:00.

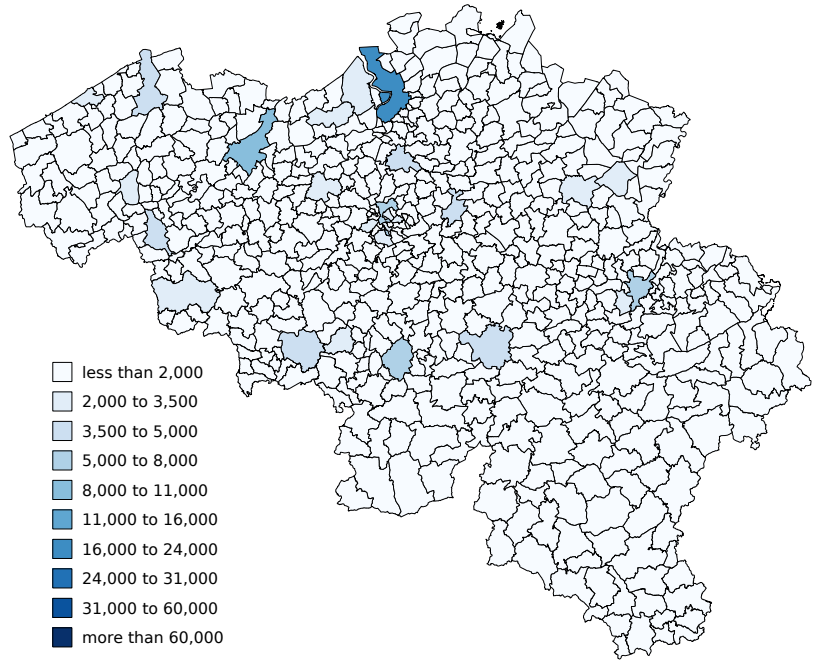

Figure 3.21 – Starting activities by municipality between 6:00 and 7:00.

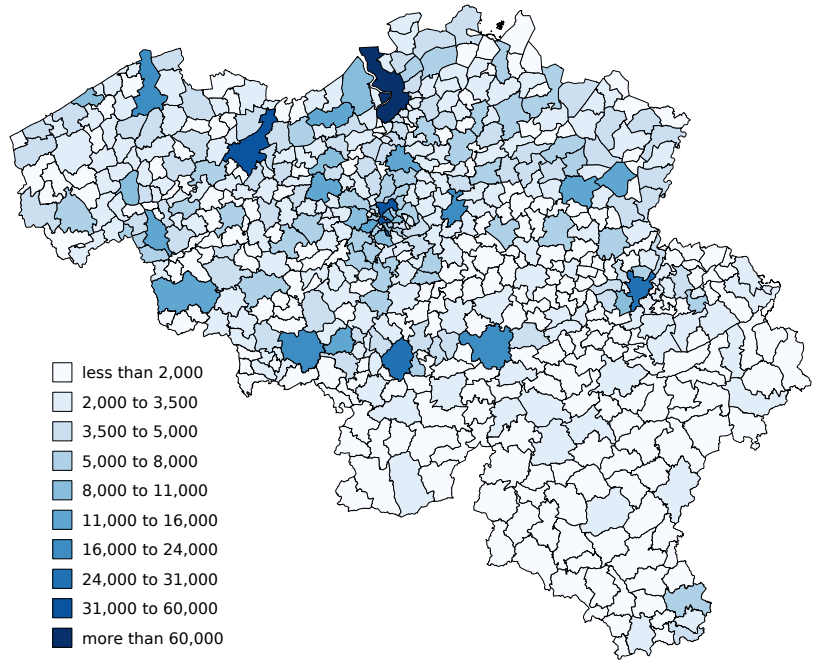

Figure 3.22 – Starting activities by municipality between 7:00 and 8:00.

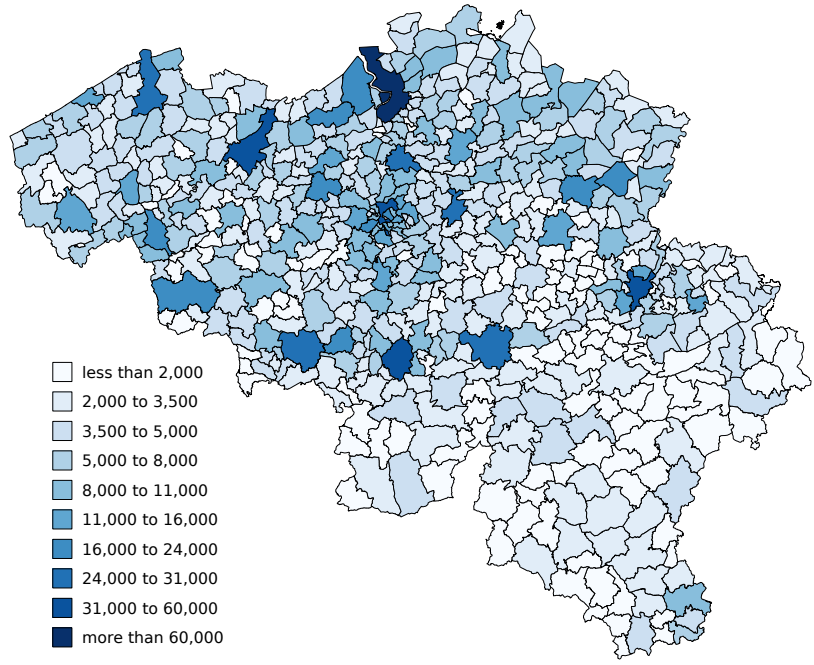

Figure 3.23 – Starting activities by municipality between 8:00 and 9:00.

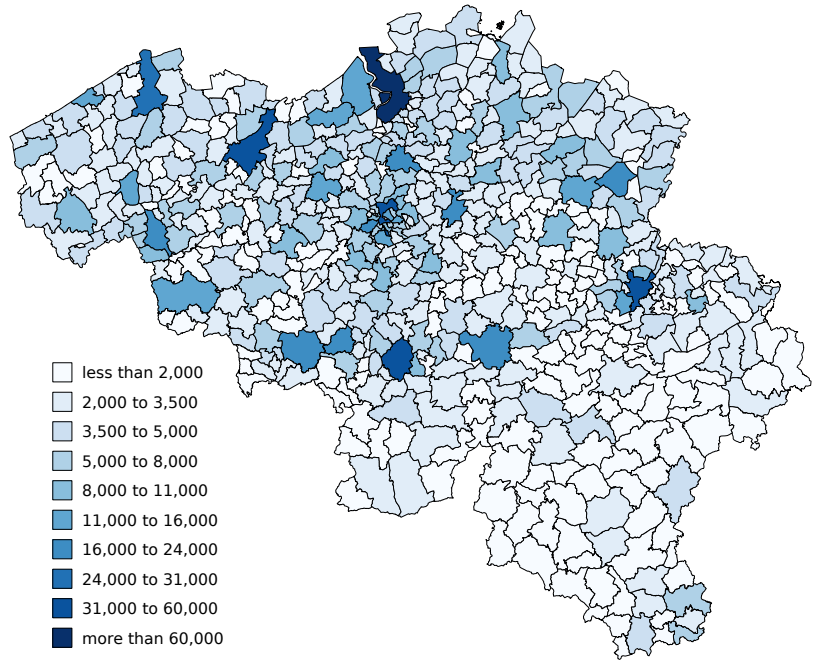

Figure 3.24 – Starting activities by municipality between 9:00 and 10:00.

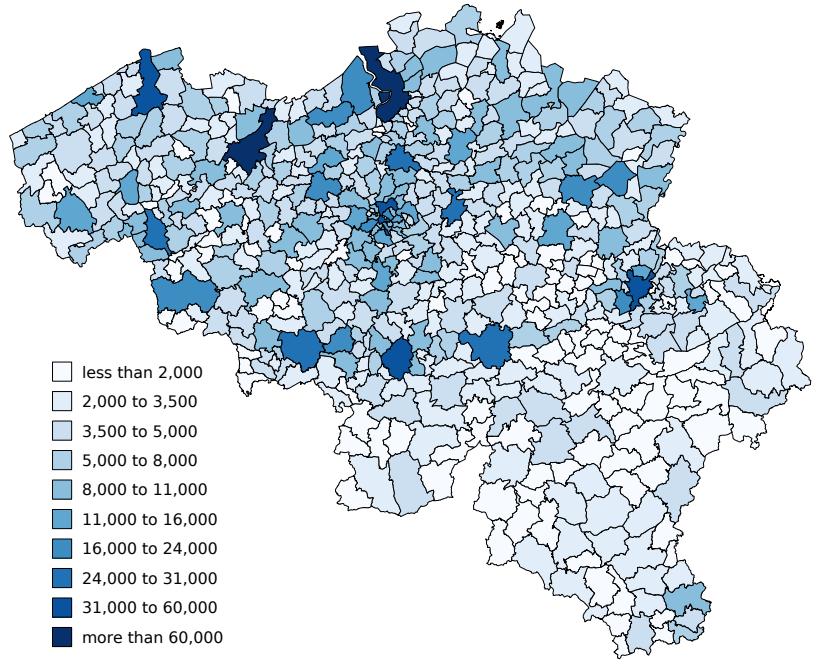

Figure 3.25 – Starting activities by municipality between 10:00 and 11:00.

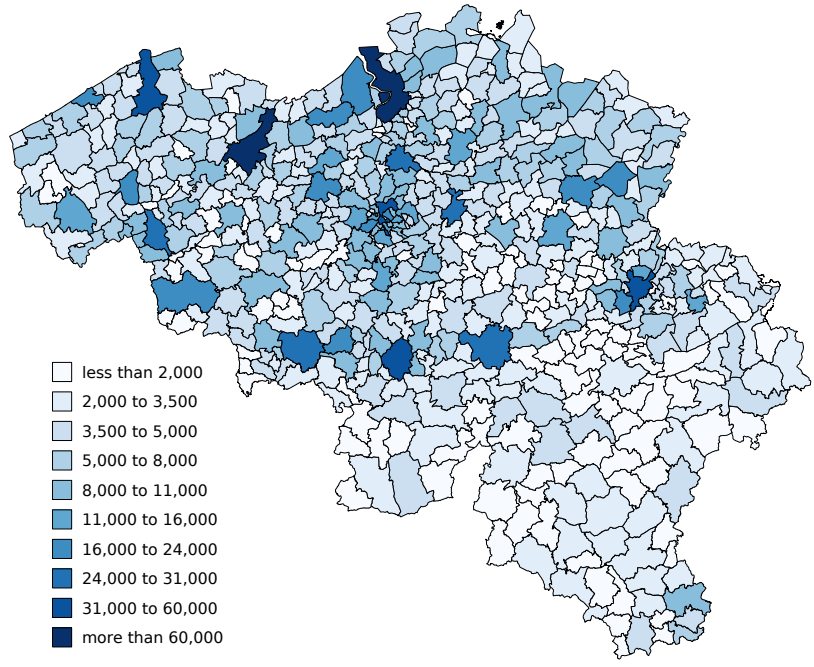

Figure 3.26 – Starting activities by municipality between 11:00 and 12:00.

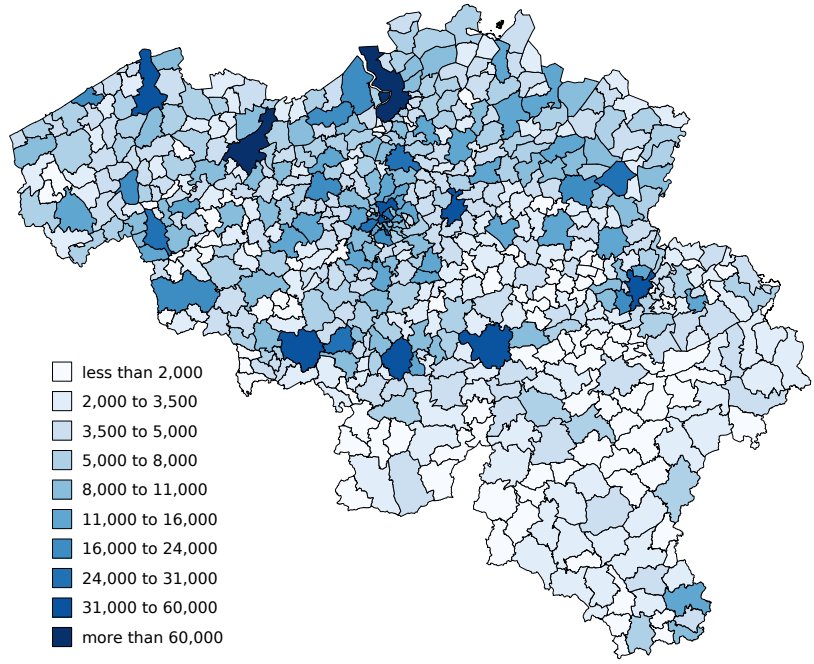

Figure 3.27 – Starting activities by municipality between 12:00 and 13:00.

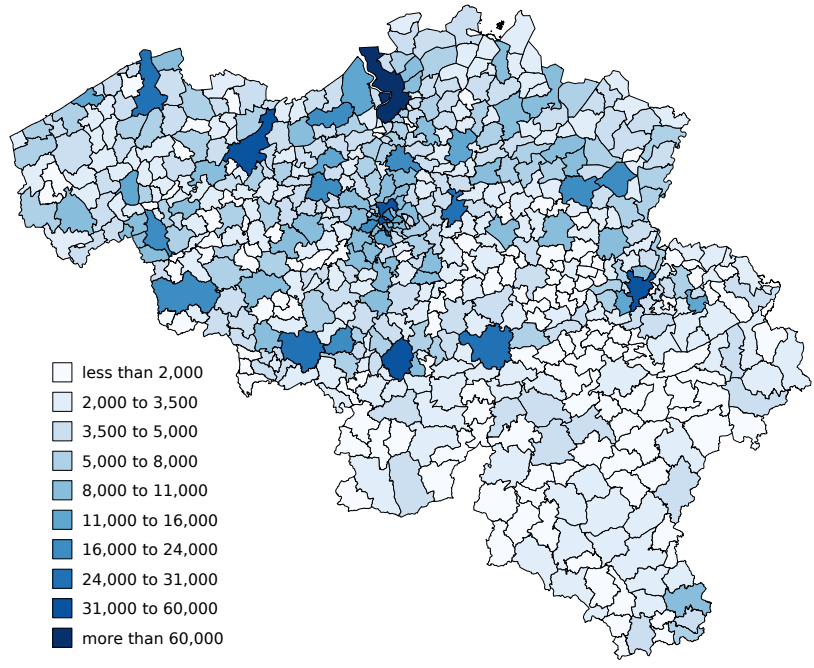

Figure 3.28 – Starting activities by municipality between 13:00 and 14:00.

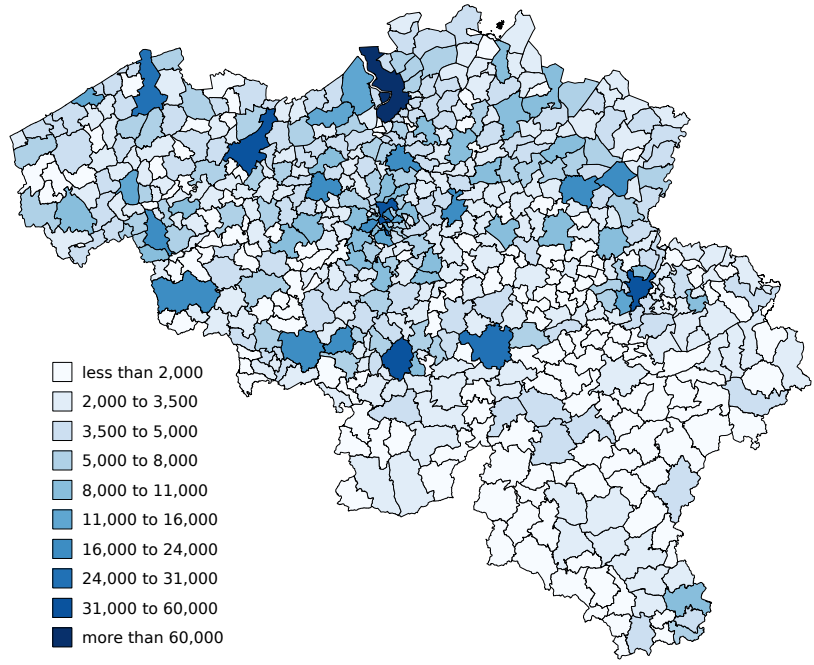

Figure 3.29 – Starting activities by municipality between 14:00 and 15:00.

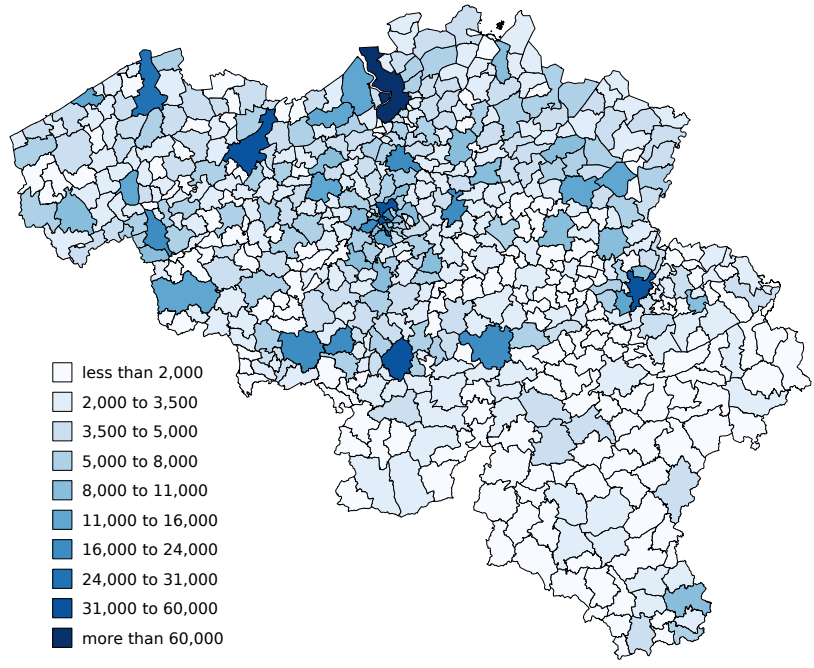

Figure 3.30 – Starting activities by municipality between 15:00 and 16:00.

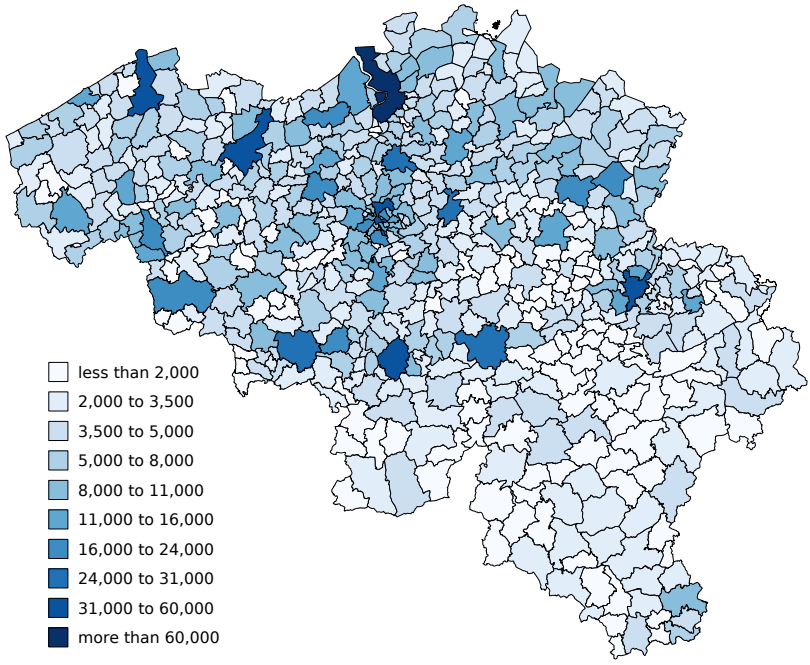

Figure 3.31 – Starting activities by municipality between 16:00 and 17:00.

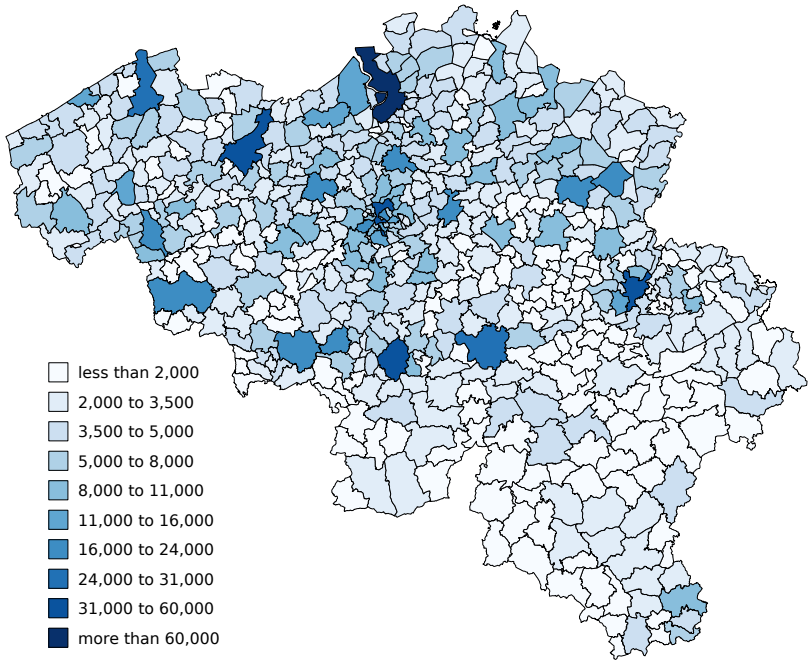

Figure 3.32 – Starting activities by municipality between 17:00 and 18:00.

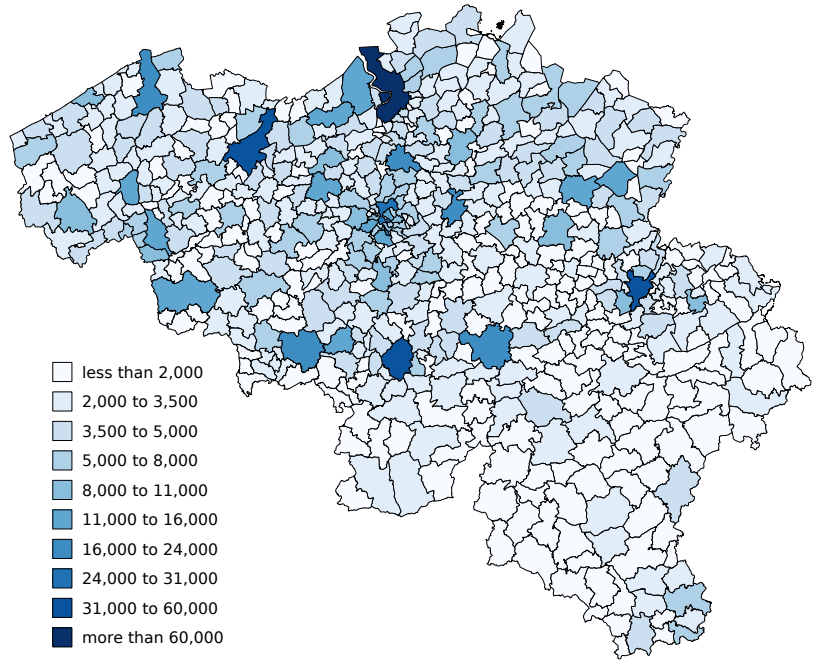

Figure 3.33 – Starting activities by municipality between 18:00 and 19:00.

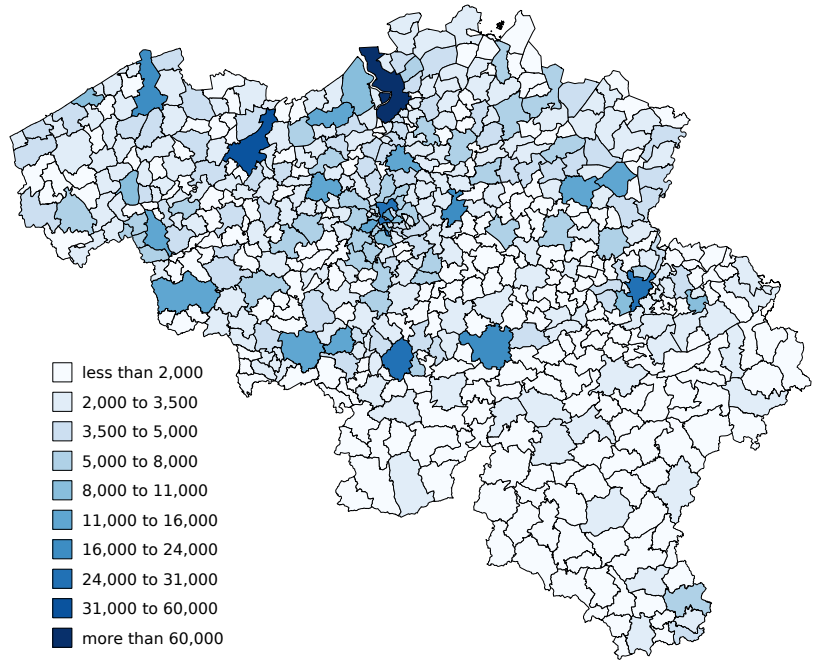

Figure 3.34 – Starting activities by municipality between 19:00 and 20:00.

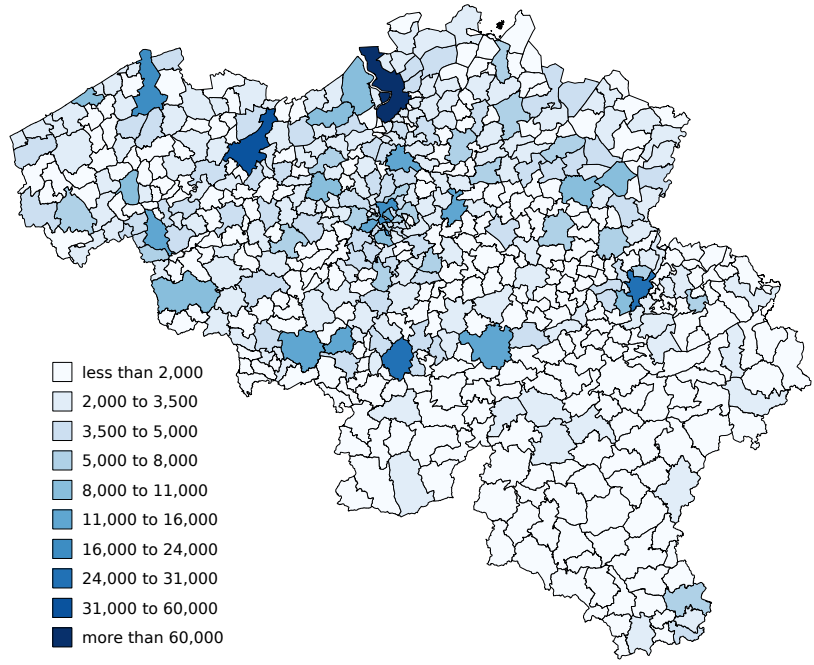

Figure 3.35 – Starting activities by municipality between 20:00 and 21:00.

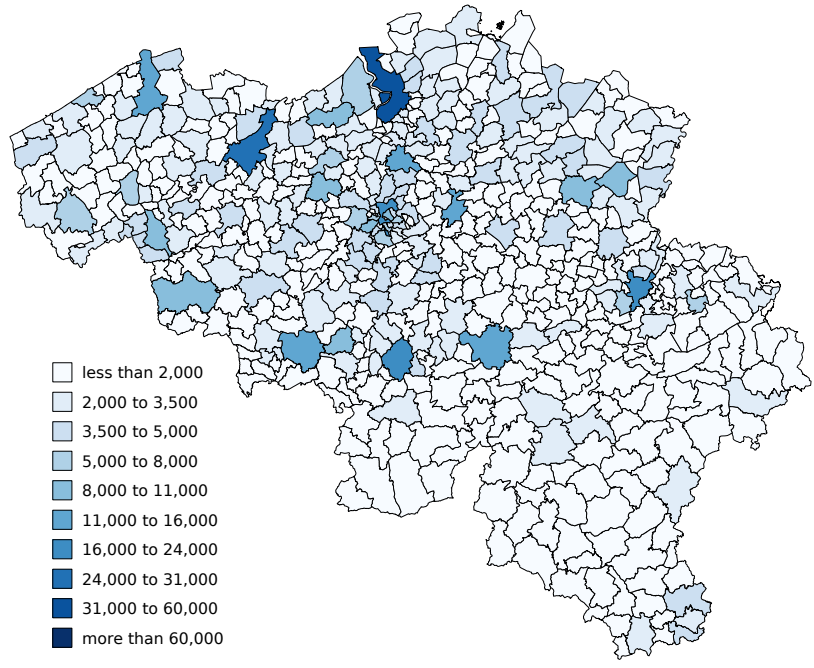

Figure 3.36 – Starting activities by municipality between 21:00 and 22:00.

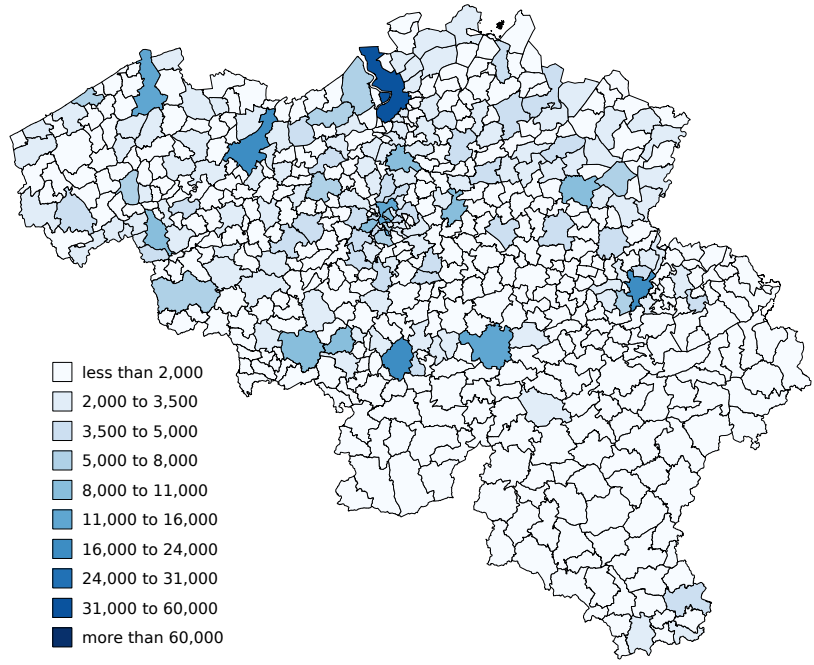

Figure 3.37 – Starting activities by municipality between 22:00 and 23:00.

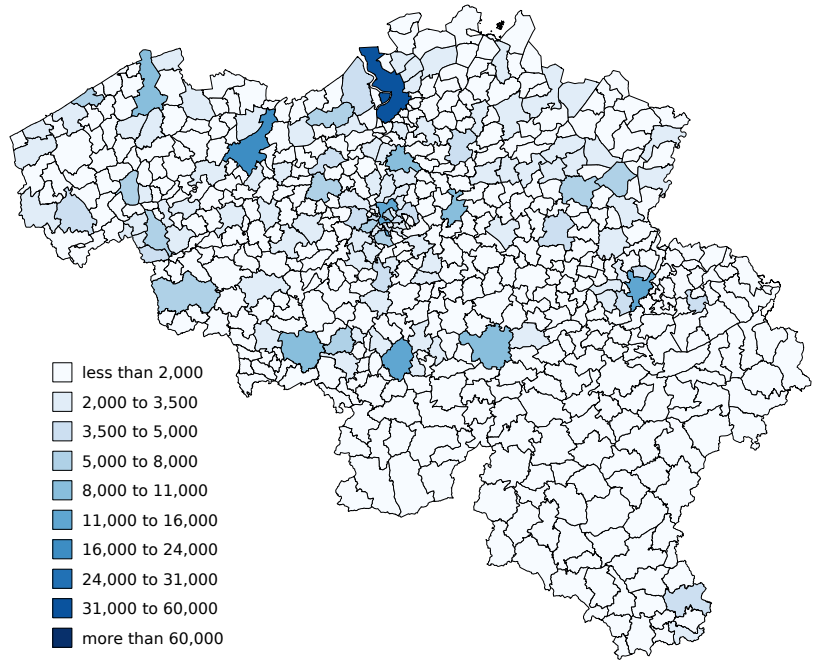

Figure 3.38 – Starting activities by municipality between 23:00 and 24:00.

As the XML output of VirtualBelgium is compatible with MATSim (see Chapter 1), it is possible to use it to perform dynamic traffic assignment. For instance Figure 3.39 illustrate a snapshot of the beginning of the morning peak on the Namur city road network. It is nevertheless important to note that every agents use the same transport mode, namely the car, as no mode choice model is currently available in VirtualBelgium.

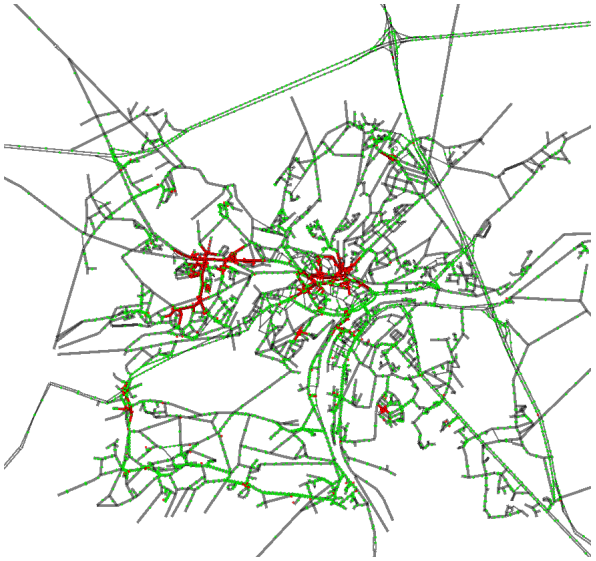

Figure 3.39 – Snapshot of Matsim output. Red agents are stuck in a traffic jam.

3.5 Conclusions

This Chapter detailed a flexible activity based model implemented in VirtualBelgium, a large agent-based micro-simulation designed to replicate the mobility behaviour of the Belgian population and its evolution. This work demonstrates that assigning and fully characterized (temporally and spatially) a sequence of activities to more than 10.000.000 agents is nowadays feasible.

The models developed in the VirtualBelgium micro-simulator are data driven and require no a priori information about the localization of activities. Indeed the minimal requirements are a road network and distributions of the distance and duration for each activity type. Nevertheless the methodology is designed to easily take advantage of any new data sources available such as precise geo-localization of schools and shopping centres, job and services indicators by municipality,*etc* in order to weight or constraints the random draws to specific nodes or municipalities. Moreover the results are promising as the

agents mobility behaviour is statistically similar to the ones observed in the Mobel mobility survey. Lastly, the outputs of VirtualBelgium are compatible with MATSim, a powerful and validated micro-simulator for traffic assignment. Nevertheless this last result must be put into perspective. Indeed MATSim is not able to simulate the 43,300,000 trips performed by whole VirtualBelgium agents on the Belgian road network, even with a powerful workstation[(4)]. It can then be used only with a sample of the population. This last observation lead us to propose in the next chapter a dynamic traffic assignment model with lower computational costs.

[(4)] equipped with two 8-cores Intel(R) Xeon(TM) E5-2660 @ 2.2GHz processors and 128Gb of RAM running a 64 bits Linux operating system

Chapter 4

Dynamic traffic assignment with strategic agents

Contents

4.1 Introduction

Traffic flows simulation is the natural follow up of travel demand forecasting and represents a central part of traffic micro-simulators such as MATSim (Meister et al., 2010), DynaMIT (Ben-Akiva et al., 1998) and AIMSUN (Barceló and Casas, 2005) as well as the traffic modelling part of UrbanSim (Waddell, 2002) and ILUTE (Salvini and Miller, 2005) integrated simulators. This component is in charge of executing the daily plans of simulated individuals in a physical environment, *i.e.* representing the traffic flows dynamics on a road network.

In recent decades, dynamics traffic assignment (DTA) models emerged for solving this problem (see Chiu et al., 2011 for an extensive description of these techniques), which either aim at reaching a steady-state of the considered system or at simulating the agents route choice behaviours. A steady-state of the system is achieved when it reaches either one of the following.

User equilibrium the journey times on all the routes actually used are equal, and less than those which would be experienced by a single vehicle (or user) on any unused route (Wardrop's first principle, Wardrop, 1952);

Stochastic user equilibrium no user believes he can improve his travel time by unilaterally changing routes (Daganzo and Sheffi, 1977).

DTA techniques can also be distinguished by their analytical or simulation-based nature. Analytical methods formulate the traffic assignment as non-linear programming and optimization problems or variational inequalities instead of focusing on the agents' behaviours. Examples of such works include Friesz et al. (1993), Merchant and Nemhauser (1978*a*) and Merchant and Nemhauser (1978*b*). Even though they have demonstrated their usefulness and are grounded on sound mathematical theories, their complexity and computational cost make their application to large-scale scenarios difficult (Peeta and Ziliaskopoulos, 2001).

Hence simulation-based methods, which explicitly model the individuals mobility behaviours, have recently gained more attention in the literature (Nagel and Flötteröd, 2009, Bazghandi, 2012 and Ben-Akiva et al., 2012). The underlying idea is to compute an user equilibrium by means of iterative simulations. These successive steps generate traffic flows until the travel time of every agent becomes stationary, *i.e.* reaches a (stochastic) user equilibrium. This class of models is more suited to an agent-based approach than the analytical ones by focusing on agents' mobility behaviour rather than optimizing a complex objective function. Nevertheless, due to their iterative nature they are also endowed with computational issues. Indeed if the road network and the number of agents involved are large, the DTA algorithms of this type may converge slowly to an equilibrium state (Pan et al. 2012).

We can observe that both categories of DTA methods for steady-state solutions are not suited to temporal networks as the agents lacks of real-time response to network modifications. For instance if an accident occurs at some

point of an agent's trip, if the number of agents in the network changes, or if the network is modified by adding/removing streets the whole optimization/iterative stages must be repeated to compute a new equilibrium. Moreover these steady-states approaches rely on strong assumptions and have several limitations now well identified. We refer the reader to Dehoux and Toint (1991) for a discussion of these limitations and why these models should be avoided in favour of purely behavioural models such as the ones proposed in PACSIM (Cornélis and Toint, 1998), FREESIM (Rathi and Nemeth, 1986) and CARSIM (Benekohal and Treiterer, 1988). The interested reader may find a recent review of these schemes in Pel et al. (2012).

In this chapter we detail an attempt for an original behavioural DTA model which is particularly appropriate in the context of agent-based microsimulation. Indeed a behaviour can be seen as one of the agent's characteristics. The proposal relies on the assumption that travellers take routing policies rather than paths (Gao et al., 2010*a*), leading us to introduce the possibility for each simulated agent to apply a strategy allowing it to possibly re-route his path depending on perceived local traffic conditions. This re-routing process allows the agents to directly react to any change in the road network, which removes the need of restarting the whole simulation process and consequently decreases the computational cost. For the sake of simplicity, we decide to model the agents' strategy with a simple neural network whose parameters are determined during a preliminary learning stage. This new approach has been applied on several networks, characterized by different congestion levels and sizes to assess its effectiveness and robustness.

This chapter is organized as follows. Section 4.2 formally details the design of the agents' strategies and their optimization process. The resulting mobility behaviour is then illustrated under various scenarios, testing the robustness of the strategies, in Section 4.3. Finally concluding remarks and perspectives are discussed in Section 4.4.

4.2 Methodology

This Section details a dynamic traffic assignment model relying on strategic agents perceiving local traffic conditions and thus adapting their behaviour to reduce their trip duration. We represent each agent as a *neural network* whose inputs are the local informations about the route network and whose output is the action to undertake: stay on the same path or modify it (keeping unchanged the destination). Because we are interested in modelling adaptive agents able to well perform in many different scenarios, we decide to introduce a learning phase instead of fine tuning by hand each agent's behaviour.

We firstly describe formally the agents' strategy based on a neural network, and its application on a road network. The remaining of the Section is then devoted to the learning process of the agents, optimizing their strategies via a genetic algorithm.

4.2.1 A neural-network based strategy for dynamic traffic assignment

Assume a road network $G = (N, L)$, where N and L correspond respectively to the sets of nodes and links which can be though respectively as crossroads or junctions and roads and a set $\mathcal{A}$ of n_{car} agents having a source node and a destination node within the network. Each link of the network is associated with a cost being the time needed by an agent to go through it. Obviously this cost is related to the capacity of the link, its free-flow speed and the number of agents already using it. The well known BPR equation from the Bureau of Public Roads (1964), originally designed for the United States highway network, formulates this relation for a link $l \in L$ as

$$S(v_l) = t_l \left(1 + 0.15 \left(\frac{v_l}{c_l} \right)^4 \right) \tag{2.1}$$

where t_l, v_l, c_l and $S(v_l)$ correspond respectively to the free-flow travel time (hours), the current volume of traffic (number of agents), the theoretical capacity (agents/hour) and the travel time on link l (hours). Figure 4.1 illustrates the speed reduction as a function resulting from equation (2.1) for a free flow speed of 50 km/h. Note that a saturation state is never reached on the link when using the BPR formulation, as the speed only reaches 0 km/h asymptotically.

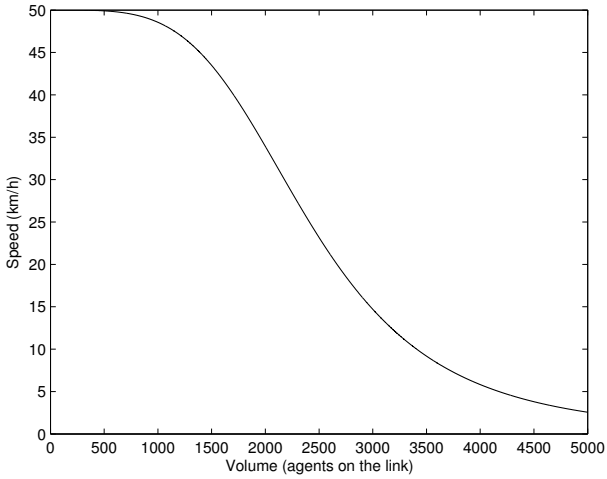

Figure 4.1 – Speed reduction with respect to the number of agents given by the BPR equation. The link considered has a free-flow speed of 50 km/h and a capacity of 1,500 agents/hour. As the number of agents on the link increases and exceeds its capacity, we observe a rapid decrease of their speed, ultimately leading to a traffic jam.

Initially each agent plans a nominal shortest path source-destination, assuming he can travel at free-flow speed, *i.e.*

$$v_l = 0 \quad \forall l \in L.$$

This shortest-path computation implies that the agents have a complete knowledge of the network topology and have a fully rational behaviour, which are strong assumptions[1] (Downs and Stea, 1977). Agents choices being independent from each other, it can result in some links being overcrowded and therefore presenting a severe speed reduction. Hence these parts of the path will no longer be the optimal ones.

As proposed by Bonsall (1992) we thus introduced a possibility for each agent to apply a strategy allowing it to re-route his path or stay on it (without modifying its source, destination and departure time) given perceived local network conditions. This strategy is realised using a neural network: a machine learning method used to solve a wide variety of tasks that are difficult to solve using ordinary rule-based programming. We refer the reader to Kriesel (2007) for a comprehensive description of these methods.

We decided to use a simple and straightforward neural network implementation, where the strategy is realised with two inputs and one output. Its simple design is presented in Figure 4.2. The input nodes x_1 and x_2 respectively reads

- the *normalised*[2] time spent from the source up to the current position;
- and the *normalised*[3] number of cars in the next link on the path.

The binary output node y_{out} gives 1 if the agent's strategy is to change its path, or 0 otherwise. For the sake of simplicity, there is no hidden layer between the input and output nodes, thus the output is given by

$$y_{out} = \mathbb{1}(w_1 x_1 + w_2 x_2 - \theta > 0) \tag{2.2}$$

where $\mathbb{1}$ is the indicator function, w_1 and w_2 are synapses weights and θ the threshold of the output node. If the agent choose the re-routing, then he computes a new shortest-path avoiding the congested link between his current location and his destination.

This strategy seems reasonably behaviourally consistent. Previous research works (Wachs, 1967, Ueberschaer, 1971 and Bonsall and May, 1986) highlighted the fact that if an agent already spent a larger amount of time en-route than it should have taken, and if he perceives congestion on the next road he intents to take, then the agent may reconsider a re-routing to avoid it.

More sophisticated neural network could be considered by adding hidden layers or inputs (Bonsall, 1992), such as the theoretical time saving the agent

[1] note that the user equilibrium models also rely on these assumptions

[2] normalised means divided by the nominal time one should have spent, *i.e.* in free flow conditions

[3] normalised means divided by the link capacity

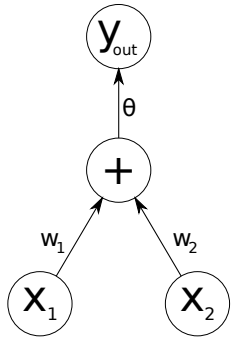

Figure 4.2 – A neural network design for strategic agents. The input layer consists of nodes x_1 and x_2 which are respectively weighted by w_1 and w_2. If their weighted combination exceeds a threshold θ then output node is activated and $y_{out} = 1$; otherwise $y_{out} = 0$. The links between nodes are called *synapses*.

would experiment if he re-routes its the initial path, the number of re-routing he already achieved, the remaining distance to its destination, the next road type and/or its length, a memory of previous choices, *etc.* Nevertheless we focused on a simple strategy in our work in order to reach a trade-off between simplicity and efficiency of the strategy.

The traffic dynamic is then performed in a synchronous way, that is all the agents $a \in \mathcal{A}$ move at the same time, each one on a given link, by an amount of space given by

$$\frac{d_{l,a}}{S(v_{l,a})}\Delta t = s_a \Delta t \tag{2.3}$$

where Δt is the time step, $S(v_{l,a})$ corresponds to the time necessary for agent a being currently on link l to reach its end, $d_{l,a}$ to the remaining distance for a to reach the end of l and s_a the agent's current speed. The time step is event-driven and determined at each iteration as the minimum time required by an agent to reach the next junction, *i.e.*

$$\Delta t = \min_{a \in \mathcal{A}} S(v_{l,a}) \tag{2.4}$$

The agent minimizing Δt has reached a node, and then computes its strategy and decides what to do. The neural network design and traffic dynamics being fixed, we can turn now to the details of the strategy optimization.

4.2.2 Strategy learning with genetic algorithm

In our context we aim to minimize the time needed to perform a given source-destination trip. Stated differently, the goal is to allow the agent to choose the best links, *i.e.* the less congested ones on his path taking into account the dynamically varying traffic conditions. As previously stated, we decided to

consider a neural network framework; the strategy parameters to be determined are then the weights (w_1, w_2) and the threshold θ that define a chromosome $\chi(w_1, w_2, \theta) \in \mathbb{R}^3$. The chromosome is associated with a fitness value $\in [0, 1]$ that reflects how optimal it is: the higher the fitness, the lower the travel time.

Note that the standard continuous optimization methods are not applicable in our context of fitness maximisation. Indeed an agent fitness depends on number of agents he encounters on each link he passes by. Hence it is a discrete optimization problem whose mathematical formulation depends on every agent's path and thus become intractable. Consequently we choose to solve this optimization problem with a standard genetic algorithm.

A genetic algorithm is a heuristic search that mimics the process of natural selection in order to find an optimal solution to a given problem. This methodology belongs to the class of evolutionary algorithms, which generate solutions using techniques inspired by natural evolution such as mutation, selection, and crossover. We refer the reader to Eiben and Smith (2003) for a detailed overview of this methodology.

First we generate an initial population of chromosomes $G_0 = (\chi_1^0, \ldots, \chi_n^0)$ where χ_i^0's components are randomly drawn from an Uniform distribution $U[-1, 1]$ and their fitness are evaluated. The following steps are then executed to iteratively generate h populations $G_1, \ldots, G_h$, at step $k \geq 1$

1. a virtual population G'_{k-1} formed by m elements $\in G_{k-1}$ is generated by repeating $m/2$ times:
   - randomly draw a couple of chromosomes (the parents), with replacement. The draws are weighted accordingly to the chromosomes' fitness values: the higher the fitness, the larger the probability to be selected[(4)];
   - the parents are crossed[(5)] with probability q_X and the offsprings added to G'_{k-1};
   - if the crossover is not performed, the parents are directly added to G'_{k-1};
2. a mutation operator[(6)] acts on all element of G'_{k-1} with probability q_M;
3. finally the next population G_k is constructed by selecting the n best fitted elements from the larger population $G_{k-1} \cup G'_{k-1}$.

The last point concerns the computation of the fitness which is achieved by repeating N_{rep} times the following scheme. Every agent is randomly assigned one origin-destination pair different from the ones assigned to it in previous

(4) this process is also known as roulette wheel selection

(5) crossing chromosomes means exchanging some of their parameters

(6) as the chromosomes' parameters are real numbers, a *continuous* mutation is adopted, *i.e.* we add to the parameter to be mutated a random number drawn from an Uniform distribution $U[-\delta, \delta]$ for a small $\delta > 0$

repetitions. One particular agent a_s is then provided with a strategy while all the remaining ones will not change their path. We then compute the nominal time needed by a_s to perform S times its source-destination trip and we divide it by the actual time needed. Assuming that $(l_1, \ldots, l_d)$ is the sequence of links covered by a_s to reach its destination, then the ratio is given by

$$f(a_s) = \sum_{i=1}^{d} \frac{t_{l_i}}{S(v_{l_i})}. \tag{2.5}$$

Finally, the average of the ratios over the N_{rep} origin-destination pairs is the fitness associated to this strategy.

Figure 4.3 illustrates the fitness function for the *2 cities* scenario detailed in Section 4.2.2.1, assuming that $w_1 = \cos\alpha$ and $w_2 = \sin\alpha$ with $\alpha \in [0, \pi]$ (for visualization purpose). It can be observed that the objective function values present a large number of local maxima, which further justifies the use of a genetic algorithm to explore the parameters space.

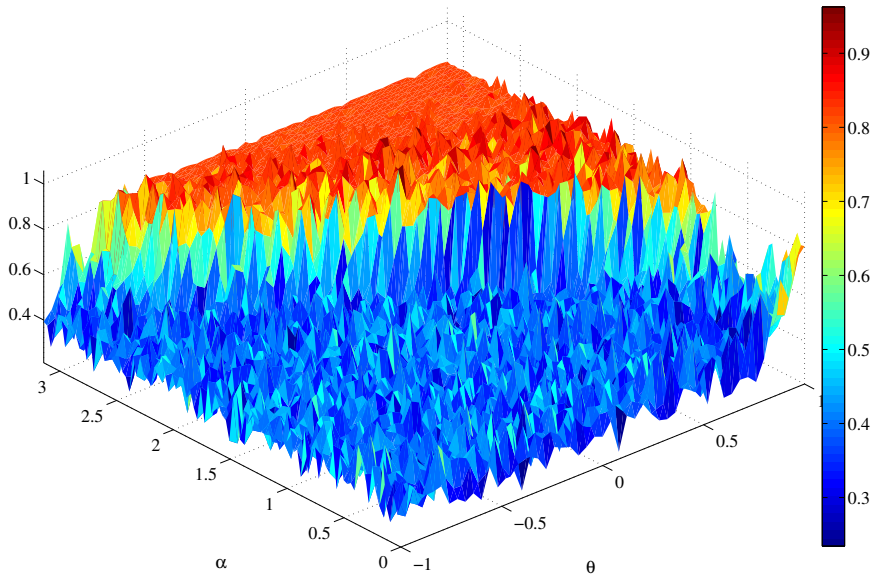

Figure 4.3 – Fitness value as a function of $\theta \in [-1, 1]$ and $\alpha \in [0, \pi]$ such that $w_1 = \cos\alpha$ and $w_2 = \sin\alpha$.

4.2.2.1 Application

The goal of the present Chapter being to present the model more than a precise application, we decided to apply this learning process with the parameters in Table 4.1 to the artificial scenario *2 cities* detailed in Table 4.2. Figure 4.4 illustrates the associated road network, which was artificially constructed for agents learning purposes and consists of 2 urban centers linked by 3 roads acting as possible bottlenecks. The origin and destination of each agents are randomly chosen among the network nodes with uniform probability.

Parameter	**Value**	**Description**
S	5	stopping criterion
n	16	population size
m	8	virtual population size
q_X	0.1	crossover probability
q_M	0.1	mutation probability
h	10	number of generations
N_{rep}	2	number of origin-destinations pairs

Table 4.1 – Genetic algorithm parameters.

	Scenario		
Parameter	2 cities	3 cities	Chicago
nodes	59	99	933
slow links (50 km/h)	182	264	354
medium links (90 km/h)	0	54	2,532
fast links (120 km/h)	0	6	64
total length (km)	196	504	13,190
network total capacity (agent/hour)	390	5,820	46,718
mean capacity per km (agent/hour)	1.9	11.5	5.7
number of agents	100	750	4,000

Table 4.2 – Scenarios characteristics.

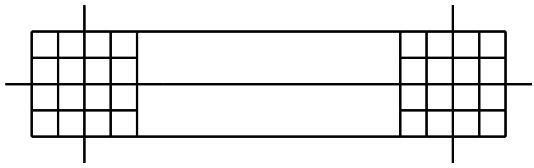

Figure 4.4 – 2 cities network.

The parameters of the genetic algorithm have been empirically determined to obtain a trade-off between the solution quality and computation times. This computation time is essentially sensitive to the N_{rep} and h parameters as the total number of iterations is given by $N_{rep} \times h$. Increasing the (virtual) population size only marginally affected the solution quality. Note that the higher the mutation and crossovers probabilities, the more the feasible solution space is explored.

We report in Figure 4.5 the fitness evolution (maximum and average values) as a function of the generation number. One can easily observes that a limited number of generation steps produces a set of strategies with high fitness, close to the optimal one. Nevertheless the number of generations has been limited to 10 since genetic algorithms are typically very computationally intensive. The current execution time is approximately 2h30 with a Matlab implementation, running on an 8-cores Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz with 16Gb of RAM.

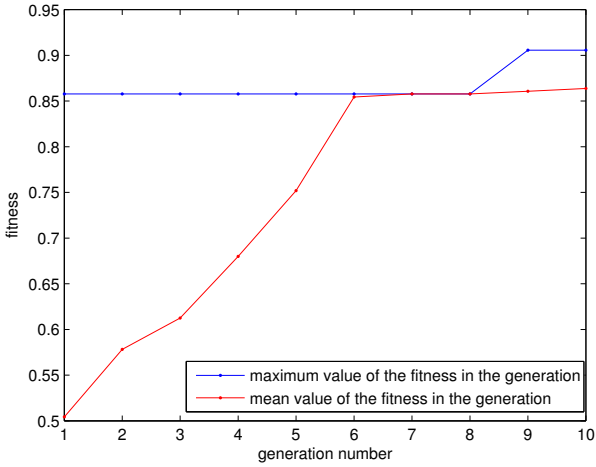

Figure 4.5 – Fitness evolution over the generations.

4.3 Results

The goal of this Section is to present some preliminary results of the routing model. In particular we are interested in analysing the robustness of strategies trained in the previous section under new environments and the impact of the proportion of strategic agents present in the population. The efficiency of the learning process is also investigated by comparing trained agents against agents following a nominal shortest path trip, that would be used as a baseline model,

and agents with random strategies[7]

We conducted experiments over 3 different scenarios detailed in Table 4.2 with different proportions of strategic agents to test their adaptability to new conditions. The scenarios attributes were chosen to explore various network conditions in terms of mean capacity per km in order to assess the strategy adaptability with respect to new environments. Each scenario is repeated twice: once with agents trained on the *2 cities* network and once with random strategic agents in order to compare the resulting traffic dynamics. In addition to the network used at the learning stage, these scenarios also involved:

- an artificial network consisting of 3 urban centres surrounded by main roads and joined by highways;
- and the Chicago road network[8];

represented in Figures 4.6 and 4.7. Note that the original capacities of the Chicago network have been downscaled to obtain congestion with less agents in order to keep reasonable computation times.

The goodness of the strategy will be evaluated using 2 indicators: the ratio of used links in the network and the fitness of the agents, *i.e.* the ratio defined by equation 2.5. The former indicator is network related: more streets used indicate a better exploitation of the network by the strategic agents as the traffic loading is more balanced over its links, thus less traffic jam are likely to occurs. The latter is dedicated to the agents' satisfaction: a high value indicates that the agent average speed is close to the one he would experience on an empty network.

Finally, the traffic flows generated by this approach will be compared with the ones determined by a more classical traffic assignment algorithm designed to compute a deterministic user equilibrium.

4.3.1 Impact of the strategic agents proportion

Let us first examine how the proportion of strategic agents in the simulation influences the average fitness of every agents. The following proportions

$$0\%, \quad 10\%, \quad 25\%, \quad 50\%, \quad 75\% \quad \text{and} \quad 100\%$$

are retained in our experiments. The evolution of the fitness is reported in Figure 4.8. One can observe that the level of strategic agents (both optimized and random) does have an impact on the average fitness experienced by the agents:

- for every scenario there is an increase of the average fitness value for proportions up to 25% optimized agents, indicating that the strategy is efficient. Moreover a proportion up to 100% of such agents have more

[7] meaning characterized by weights and threshold randomly drawn

[8] available at `http://www.bgu.ac.il/~bargera/tntp/`

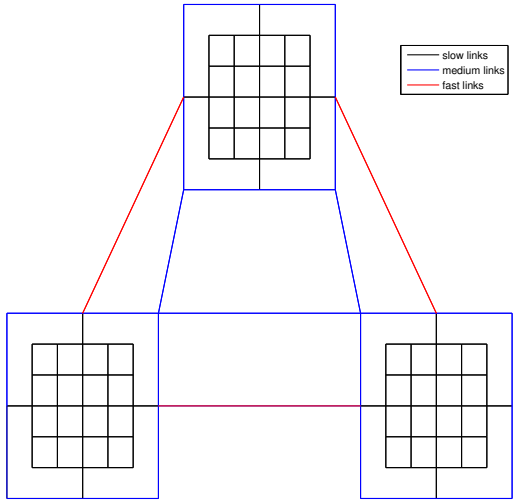

Figure 4.6 – 3 cities network.

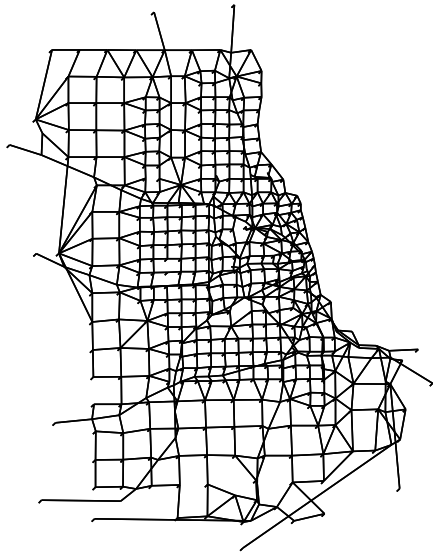

Figure 4.7 – Chicago network.

significant and positive impact in the scenarios involving uncongested road networks, *i.e.* with higher mean capacities (*3 cities* and *Chicago*). For congested network (*2 cities*), if there are too many optimized agents, then the overall performance may not increase and even drop. Indeed when they re-route themselves, they encounter congested links again, resulting in a lower fitness as the agents always suffer a speed reduction;

- on the other hand, agents performing random re-routing experience a decrease of the overall performance as their proportion increases with the exception of the *3 cities* scenario. This behaviour is certainly imputable to the uncongested nature of the network.

Moreover the trained agents always perform better than the random agents in term of average fitness, demonstrating that the learning process is necessary and produces efficient strategies.

Similar behavioural patterns can be observed for the ratio of used streets with respect to the proportion of strategics agents depicted in Figure 4.9. As previously observed, a proportion of 25% optimized agents seems to be a good trade-off. Again agents with random strategies performed worse than agents with optimized strategies for a majority of the conducted experiments.

These observations show that the provided strategy, optimized with a preliminary learning process, is effective compared to random behaviours and even having no strategy. Moreover these findings hold even if the learning phase has been performed on a small network.

4.3.2 Performance profiles

In order to have a deeper insight of the agents' behaviours over the simulation, we propose a performance profile with 4 indicators:

- the ratio of used links with respect to the total number of streets;
- the ratio of jammed links[(9)];
- the distribution of the average fitness across the simulated agents;
- and the average fitness evolution.

Note that the simulation runs until half of the agents have performed 5 round-trips between their respective origin-destination. This is just a way to avoid cases when a single agent could have covered his path very fast by chance. On the other hand this avoids too long simulations where extreme events of very long travel times can arise. Figures 4.11 to 4.15 illustrate these profiles for every scenario once we consider including a proportion of 25% optimized and random agents.

[(9)] a link is said to be jammed if its speed computed by the BPR equation is reduced at least by half with respect to its free flow value

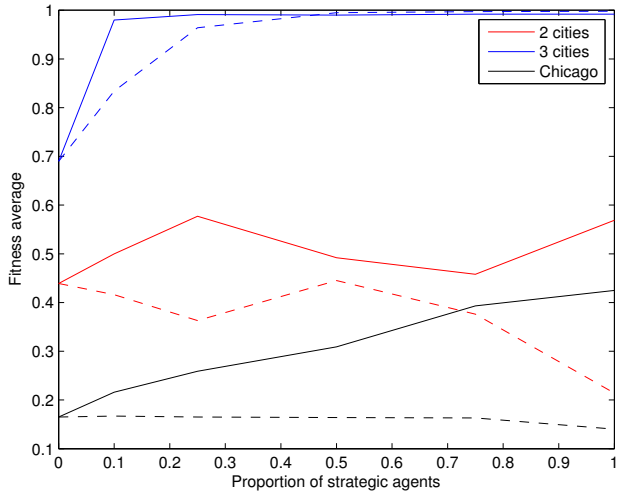

Figure 4.8 – Evolution of the agent's average fitness with respect to the proportion of strategic agents in various scenarios. The average fitness is computed at the end of each simulation. The solid lines represent the agents provided with a strategy optimized by a genetic algorithm while the dashed lines correspond to agents with random strategies.

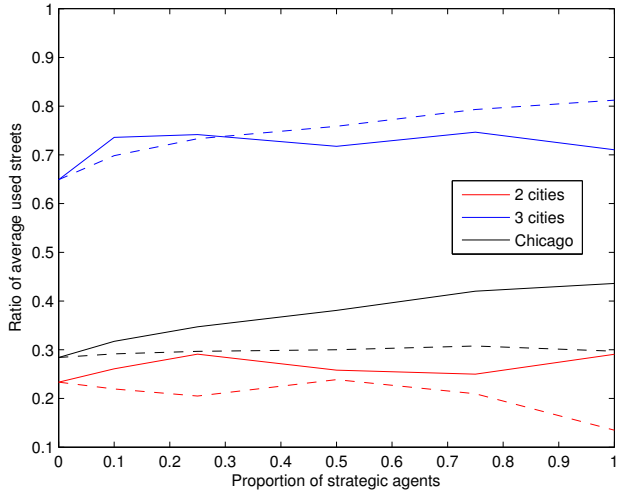

Figure 4.9 – Evolution of the average ratio of used streets over the total number of streets with respect to the proportion of strategic agents. The average ratio is computed across the whole simulation. The solid lines represent the agents provided with a strategy optimized by a genetic algorithm while the dashed lines correspond to agents with random strategies.

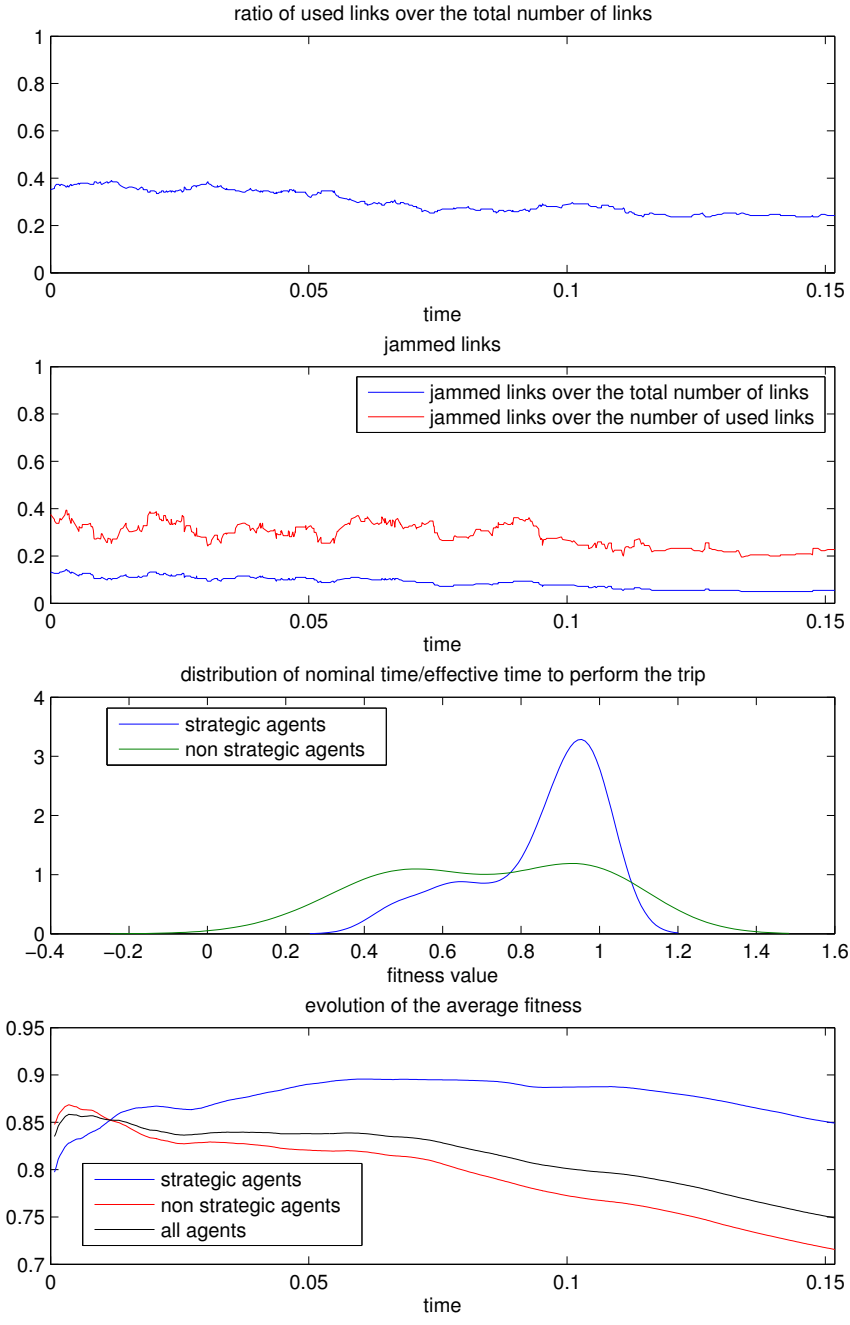

Figure 4.10 – Performance profile for scenario 2 cities - strategic agents.

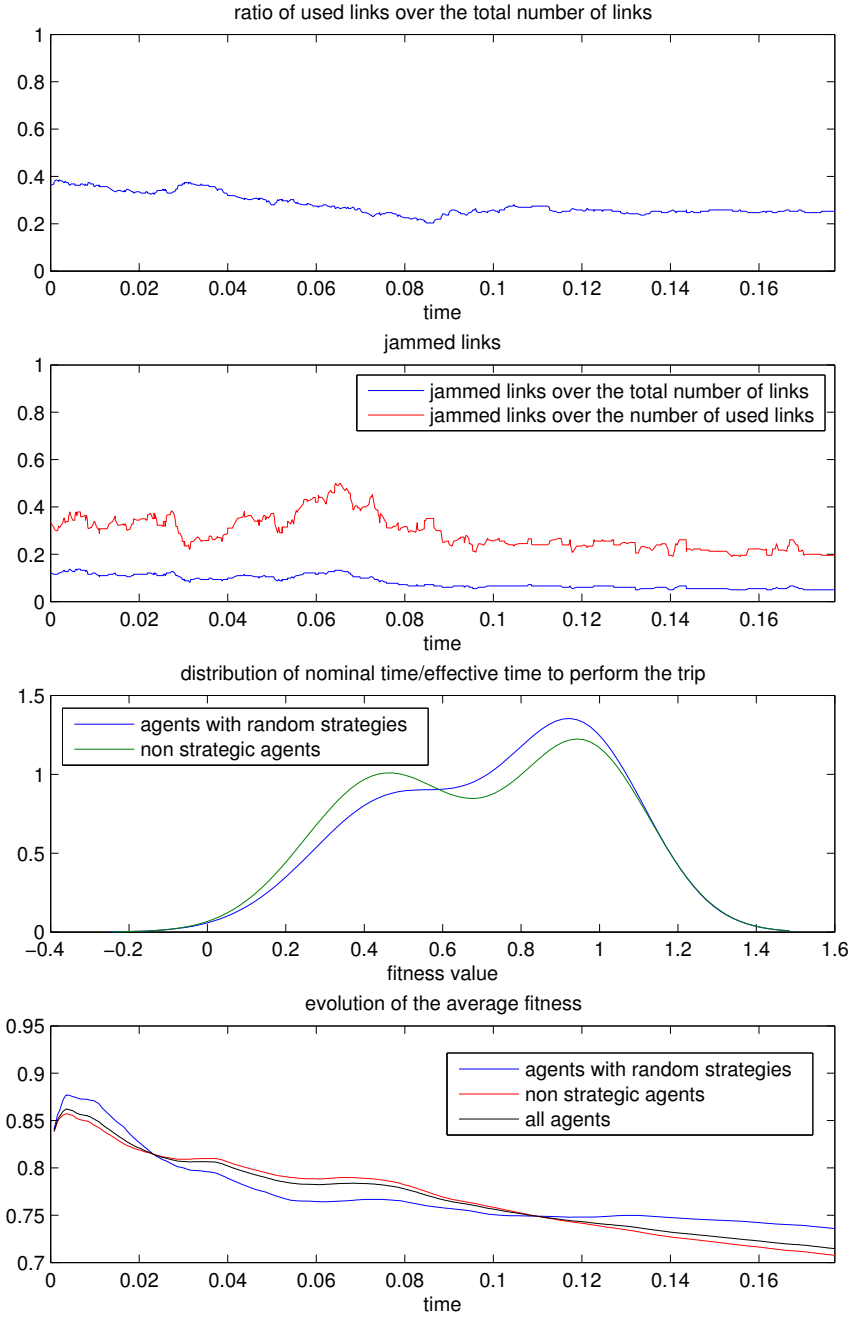

Figure 4.11 – Performance profile for scenario 2 cities - agents with random strategies.

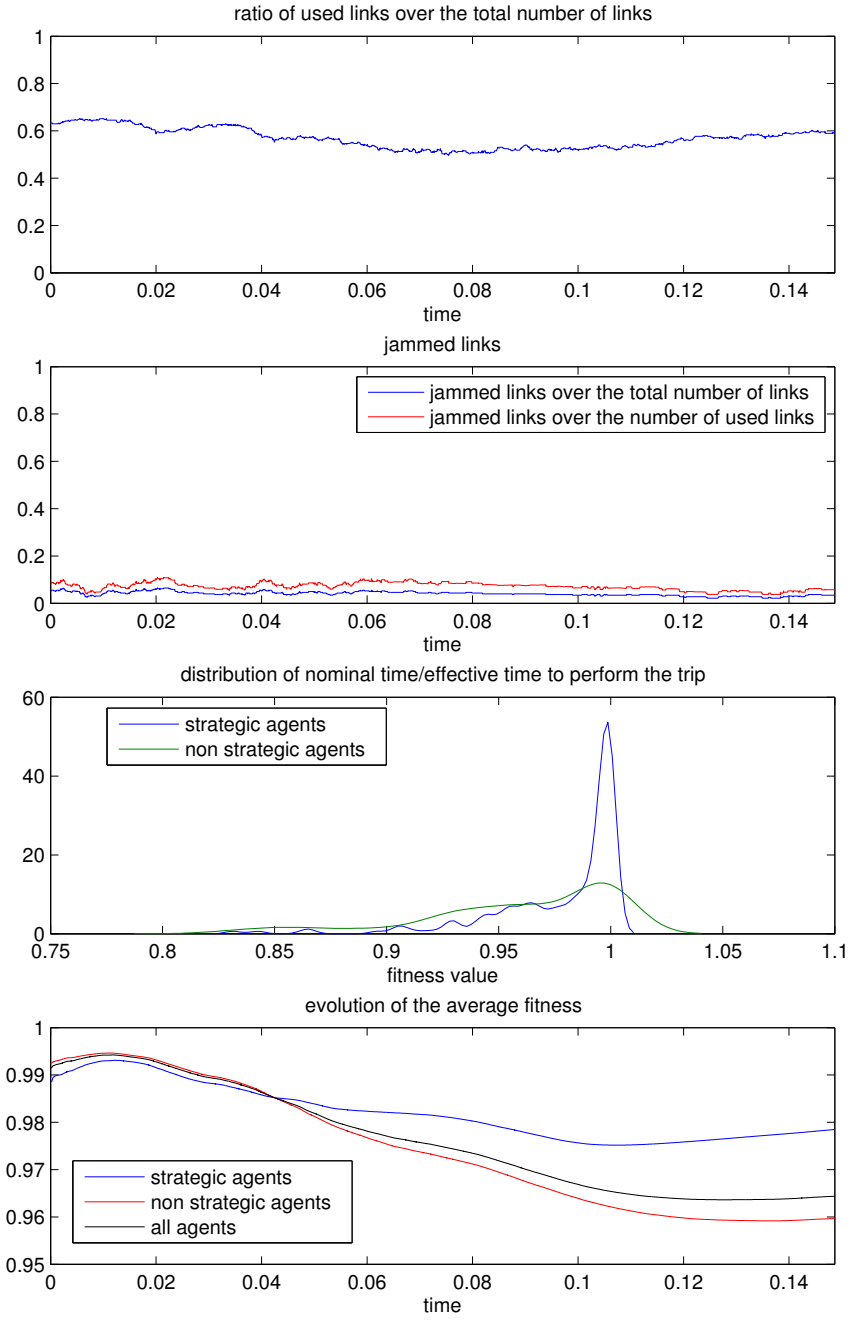

Figure 4.12 – Performance profile for scenario 3 cities - strategic agents.

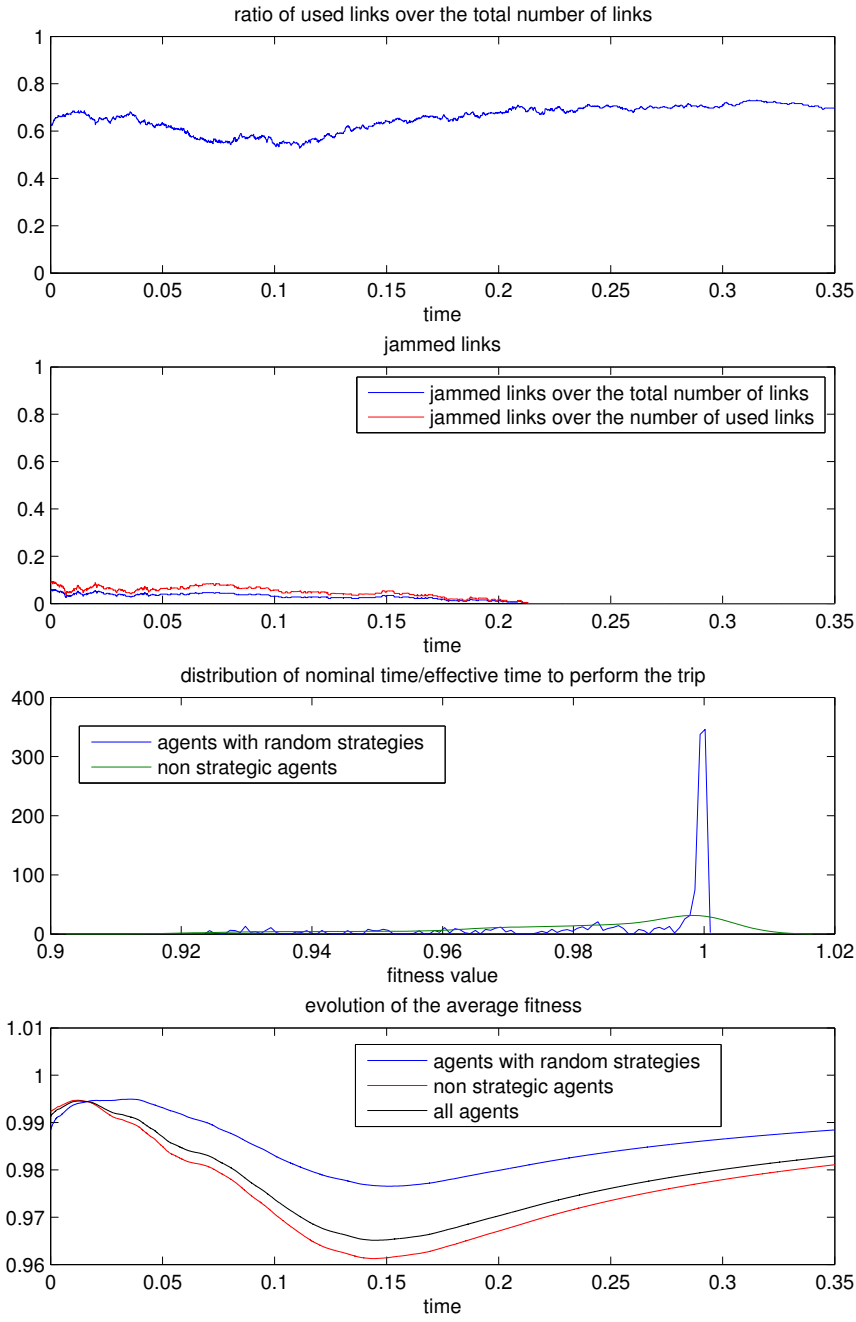

Figure 4.13 – Performance profile for scenario 3 cities - agents with random strategies.

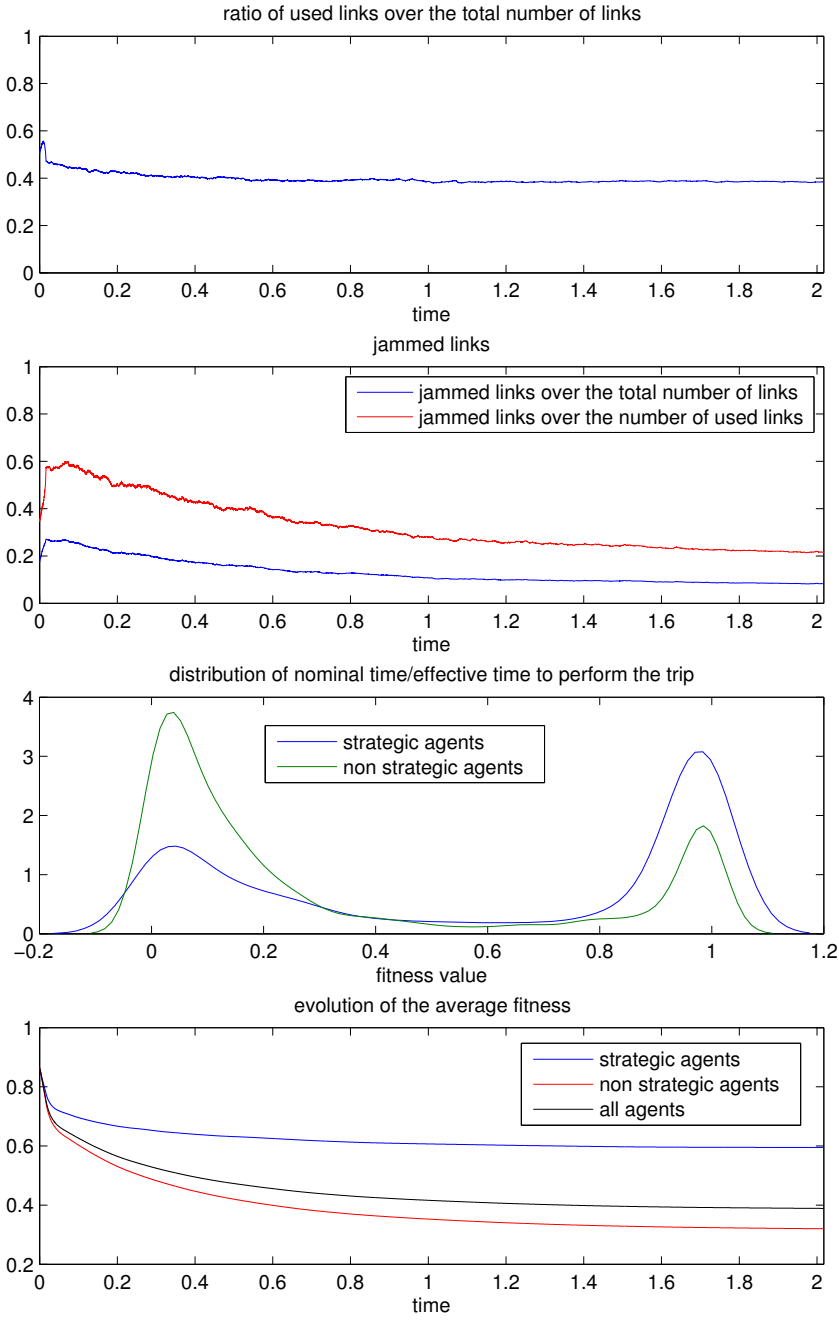

Figure 4.14 – Performance profile for scenario Chicago - strategic agents.

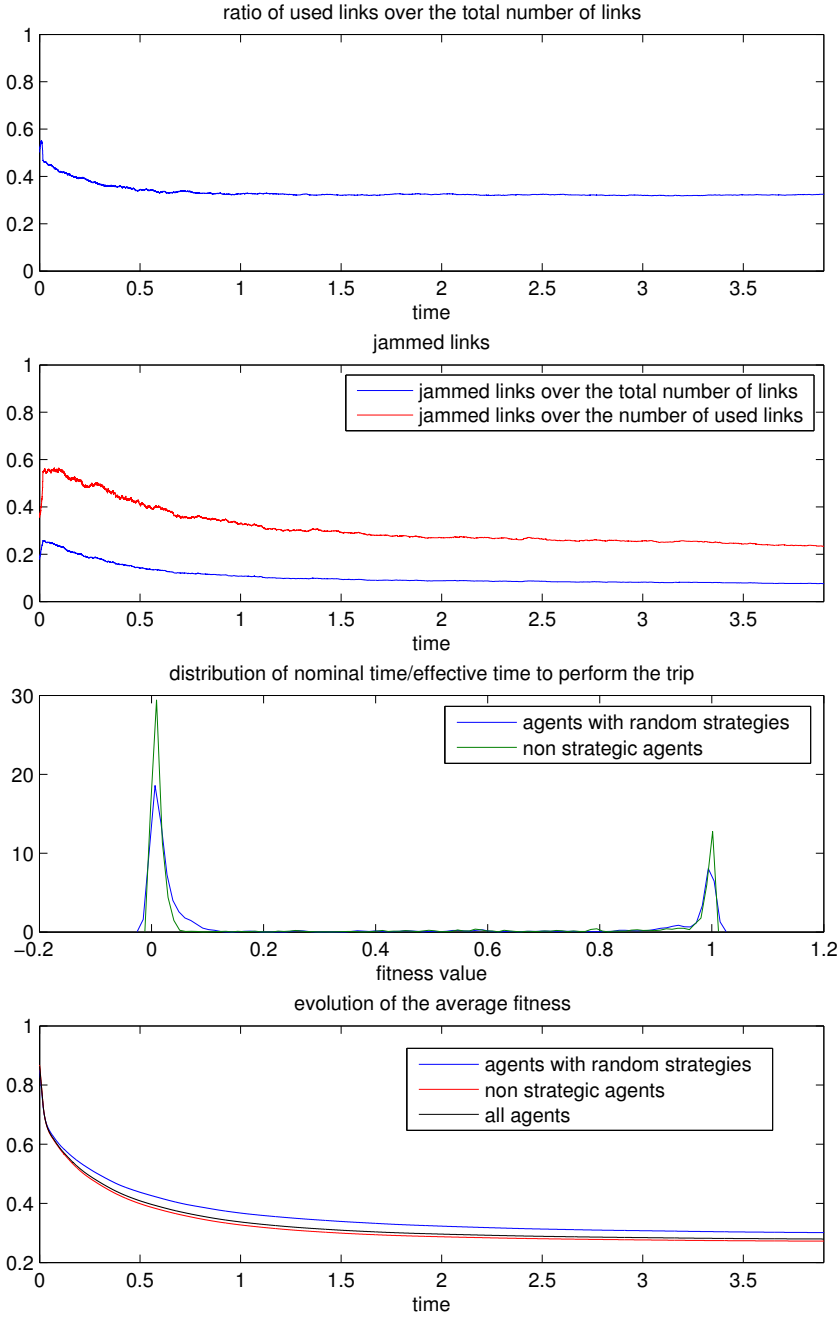

Figure 4.15 – Performance profile for scenario Chicago - agents with random strategies.

From these profiles, and more specifically from the fitness value distribution and evolution over time, it appears that the trained agents perform better that the non-strategic and random ones.

Regarding the agents' behaviour with respect to the level of traffic density, the observations made previously are also confirmed:

- in presence of high congestion, trained strategic agents have at least a fitness as good as the one associated with the other agents;
- similarly in uncongested conditions, strategic agents are able to better take advantage of the network and their fitness is significantly superior than the fitness of the non-strategic agents;
- generally the random agents have a fitness similar to the one of the non-strategic agents except in uncongested network where they have slightly better fitness.

Note that the ratios of used links and jammed links seems to rapidly converge to an equilibrium in every simulation. The initially observed decrease corresponds to the time needed for every agent to enter the simulation, namely a sort of transient time in the simulation. As the number of agents on the network remains constant, this observation comes as no surprise.

4.3.3 Agents' robustness to network modifications

We now examine the robustness of the agents' performances against network modification. This has been tested by sequentially taking out links from the *3 cities* network. Firstly we removed the fast links between the 3 cities; the following removals were done randomly and in a way to keep the network's nodes connected(10). The final network is given in Figure 4.16.

The results of these experiments, involving a proportion of 25% strategic agents, are illustrated in Figure 4.17. It can be observed that the proportion of removed links does not affect too much the proportion of used streets. On the other hand, as the network is being degraded, the strategic agents fitness median value remains higher than the one associated with the non-strategic agents, indicating that the strategic agents can adapt their routes to cope with the modifications.

4.3.4 Comparison with an user-equilibrium approach

In order to assess the validity of the proposed approach, we can compare its results to the ones generated by the Origin-based assignment (OBA) algorithm developed by Bar-Gera (2002). This algorithm determines the classical deterministic user equilibrium as defined by Wardrop (1952). The retained network

(10) in order to keep always at least one feasible path between any origin-destination pair

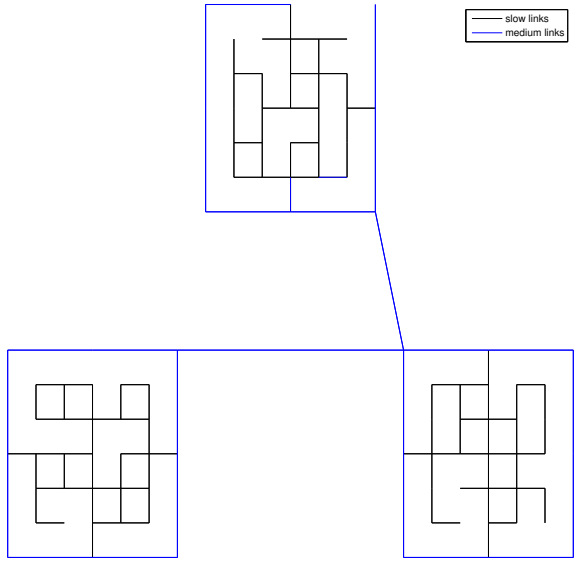

Figure 4.16 – Road network of *3 cities* after the removal of 20% of its links.

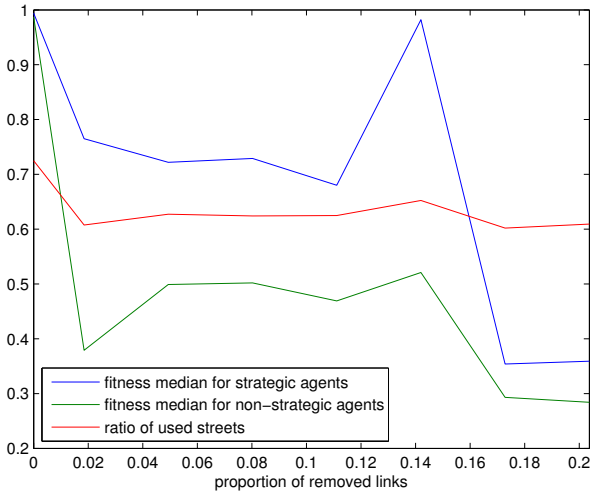

Figure 4.17 – Evolution of fitness median for strategic and non strategic agents and the ratio of used streets as a function of the proportion of removed links from the original *3 cities* network. A proportion of 25% strategic agents was used in the conducted experiments.

for these experiments is the well-known Sioux-Falls network[11] represented in Figure 4.18 and detailed in Appendix C. Although this network is not considered to be a realistic one, it was used in many publications for testing and benchmarking new methodologies.

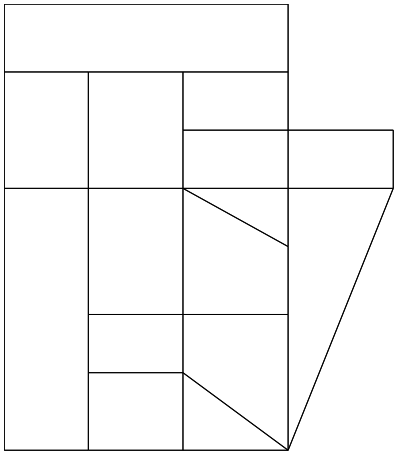

Figure 4.18 – Sioux-Falls road network: 24 nodes, 76 links (60 km/h) and 360600 trips.

Figure 4.19 shows the impact of the proportion of strategic agents on various indicators. It can be observed that

- as the proportion of strategic agents increases, the average fitness, the ratio of used streets and the agents average speed slightly improve;
- and the number of time steps required for every agents to reach its destination also decreases as the proportion of strategic agent increase up to 75%.

The comparison is performed by computing the average and maximum absolute deviations between the traffic flows produced by our approach and the ones determined by the OBA algorithm. These deviations, respectively denoted by D_a and D_m, are formally defined by

$$D_a = \sum_{l \in L} \frac{|v_l - v_l^*|}{l_{tot}} \tag{3.6}$$

and

$$D_m = \max_{l \in L} |v_l - v_l^*| \tag{3.7}$$

[11] available at `http://www.bgu.ac.il/~bargera/tntp/`

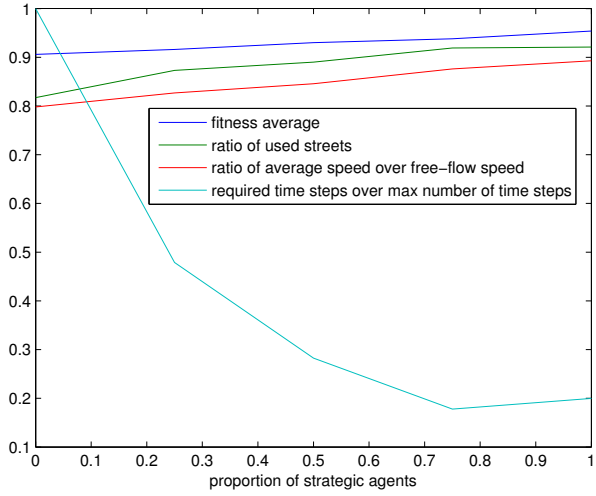

Figure 4.19 – Evolution of some performance indicators with respect to the proportion of strategic agents in the Sioux-Falls network.

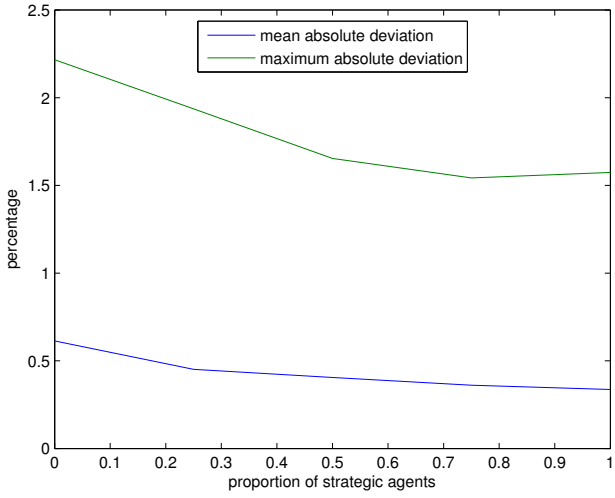

Figure 4.20 – Average and maximum absolute deviations (in percent) between theoretical flows computed by the Origin-based assignment algorithm and the proposed approach as a function of the proportion of strategic agents.

where L is the set containing the l_{tot} network links, v_l the number of agents going through $l \in L$ during the simulation and v_l^* the traffic flow computed by the OBA algorithm for the same link.

Figure 4.20 illustrates the decreases of theses deviations as a function of the proportion of strategic agents. Indeed for a proportion of 75%, D_a is less than 0.5% while D_m is about 1.6%, showing that the solution of the proposed approach is close to the theoretical one. Hence the solution proposed by the genetic algorithm is very close to the optimal one.

4.4 Conclusions

In this chapter we presented an alternative to existing simulation-based dynamic traffic assignments models. The proposal main characteristic consists in a strategy provided to the travelling agents. The strategic reactive agents examine in real time the possibility to re-route their path at any link intersection they encounter to take care of instantaneous changes in the road network. The proposed strategy consists of a neural network whose input layer relies only on current trip duration and perceived traffic conditions in the next link on his path. The neural network is trained on a simple congested network with a genetic algorithm to derive its optimal parameters. The trained agents can then use this strategy to face changing conditions. Moreover as the agents use only local information, the overall network topology does not really matter, thus the strategy should be able to cope with (larger) networks different from the one used for training.

In the conducted experiments, the use of strategic agents has demonstrated an efficient behaviour in terms of network use despite its simplicity. Their robustness and adaptability to new environments, *i.e.* different networks and congestion levels have also been demonstrated and thus indicate promising results. A key advantage of this approach is that it does not require several computationally expensive iterations to take account of network modifications or changing traffic conditions. Indeed the neural network need to be trained only once during a preliminary learning stage. As a result this model seems well-suited to large-scale applications.

Conclusions and further perspectives

Oh, I survived. Brilliant. I love it when I do that.

– Dr Who

This thesis aimed at developing VirtualBelgium, an agent-based and open-source platform for Belgian population simulation whose architecture has been introduced in the first Chapter. More specifically we have focused on the agents generation and the traffic simulation parts of the platform. The implementation of the core foundations of VirtualBelgium and its traffic simulation module represent major contributions of this work. In the following paragraphs we summarize our developments and suggest some possible directions for future research.

A micro-simulation is dedicated to simulate the behaviour of a population at the individual level to capture emergent phenomena. Hence micro-simulation requires disaggregate data on the agents of interest which is often unavailable due to privacy and/or cost reasons. This observation leads us in the second chapter to develop a synthetic population generator whose main characteristics are its sample-free nature, its ability to cope with moderate data inconsistencies and different levels of aggregation. The new algorithm has been applied to generate a synthetic population for Belgium in 2001 of 10.300.000 individuals gathered in 4.300.000 households. The conducted validation tests have demonstrated that our generator is able to produce reliable synthetic data which is crucial since micro-simulations are sensitive to it. Nevertheless, investigating the temporal evolution of the input database and the synthetic population are interesting research perspectives. Moreover adding new attributes to the agents or refining existing ones (such as the age classes) could improve the VirtualBelgium models.

In the third chapter, we designed a stochastic and flexible activity-based model for traffic demand forecasting relying on weak data requirements: only a road network and various statistical distributions are needed. Furthermore

no a priori information about the localization of activities is necessary, but its flexible design easily takes advantage of any new data sources about these localizations. Despite its simplicity, the model has demonstrated promising results as the agents mobility behaviours are statistically similar to the ones observed in the Mobel mobility survey. Future research works to be conducted for increasing the quality and the reliability of the results include :

- considering a true mode choice for reaching an activity (public transportation, car, walking);
- collecting data for destination choice (job and service indicators by municipality, schools localization, land use...);
- further refining the statistical distributions and their correlations used for determining the activities attributes;
- validating the produced origin-destination matrix by comparing it against real data (to be collected).

A traffic simulation is completed by considering the assignment of the generated demand on the road network. For that purpose, VirtualBelgium can be coupled with the MATSim framework, but in its current implementation this framework was not able to cope with the 10,000,000 synthetic agents due to the problem size. An alternative to the conventional simulation-based dynamic traffic assignment models is proposed in the fourth chapter. The proposal main characteristic is the neural network-based strategy provided with the agents, giving them the ability to re-route themselves accordingly to perceived local traffic conditions. Even tough the results tend to indicate that the strategy is efficient, only an exploratory work has been conducted. Hence it still requires further investigations and comparisons with other validated approaches are needed before considering its implementation within VirtualBelgium. Different specifications of the strategy and the speed reduction function as well as a sensitivity analysis of its learning process (by varying the network and/or the number of strategic agents) could also be considered in the future.

In conclusion it is our hope that the work described in this thesis will be useful in future developments of VirtualBelgium as an integrated microsimulator and opens new research perspectives. Indeed this research work is, we think, just the start of a powerful prospective tool, and a lot of future works are possible. For instance VirtualBelgium can provide a useful framework to the research topics developed at the Namur Center for Complex Systems (NAXYS) such as opinion and disease propagation, social network dynamics, socio-demographic evolution of a population, residential mobility, family and social dynamics, *etc.*

List of Tables

List of Figures

Appendix

Appendix A

VirtualBelgium 1.0 installation and user guide

A.1 Introduction

VirtualBelgium is an open-source project which aims at developing understanding of the evolution of the Belgian population using micro-simulation and considers various aspects of this evolution (demographics, residential choice, activity patterns, mobility, ...). This document describes how to install and use VirtualBelgium 1.0 on a 64 bits GNU/Linux operating system.

A.2 Download and directory listing

The project files are hosted at

`http://sourceforge.net/projects/virtualbelgium/files/`

and are organised as follow.

./	root directory, contains running scripts and a Makefile
./bin	VirtualBelgium executable and configuration files
./data	inputs
./doc	documentation
./include	header files
./licenses	licenses of Repast HPC, tinyxml2 and VirtualBelgium
./logs	log files of simulation runs
./outputs	outputs generated by simulation runs
./scripts	scripts for processing outputs
./tools	tools for processing outputs
./src	source files and a Makefile

A.3 Requirements

The minimal requirement to compile, run the simulation and produce outputs consists of the Repast HPC 2.0 framework. The raw outputs can then be processed trough provided scripts which also have specific requirements. This Section lists the mandatory and optional requirements.

A.3.1 Mandatory

C++ compiler

To build the project you must first have a C++ compiler installed. A common C++ compiler is *g++*; you can determine if you have g++ installed by typing:

g++ -v

If g++ is installed and is on the execution path, this will give you version information.

Make utility

The *make* utility manages dependencies and automates the compiling process. To confirm that make is installed on your system and is on the execute path, navigate to an empty directory and type:

make

If a message like

make: *** No targets specified and no makefile found. Stop.

is the response, then make is installed; otherwise install it. The make utility executes rules defined in a Makefile file.

MPI environment

The Message Passing Interface (MPI) is a standard allowing several processes to communicate together. MPI's goals are high performance, scalability, and portability. This standard remains the dominant model used in high- performance computing today. In order to check if a MPI compiler is installed, verify that the command

mpicxx

is not returning a message like

bash: mpicxx: unknown command...

Similarly, try to invoke the command

mpirun

to assess that an execution environment is already set up. If you do not have an MPI implementation installed, we recommend the use of OpenMPI.

Libraries

The following libraries need to be installed before compiling Repast HPC 1.0.1:

- libcurl ;
- Boost 1.48 (or higher): alongside with the Boost header-only libraries, it is also necessary to have the boost-mpi, boost-system, boost-serialization and boost-filesystem compiled libraries;
- NetCDF 4.2.1;
- NetCDF C++ 4.2.

Check your system documentation to install the required dependencies.

Repast HPC 2.0

Repast for High Performance Computing is a lean and expert-focused C++-based modeling system that is designed for use on large computing clusters and supercomputers. This framework is developed by Argonne National Laboratory and freely available at `http://repast.sourceforge.net`. VirtualBelgium relies on the 2.0 version. Before installing Repast HPC, make sure that the previous mandatory dependencies are installed. In order to setup Repast HPC, download it, navigate to the directory containing the downloaded archive, unzip it and follows the instructions given in the file INSTALL.txt.

If the *make* process generates an error, it is likely that the files

./src/repast_hpc/DirectedVertex.h and
./src/repast_hpc/UndirectedVertex.h

must be modified by replacing every occurrences of

```
getItems(...)
```

by

```
this->getItems(...)
```

before running the *make* utility again. If an error still occurs, then the *Makefile* file should be edited in order to remove every references to *static* compilations.

A.3.2 Optionnal

The next utilities are not required to compile and execute VirtualBelgium, but they are necessary to help debugging, generate the documentation or even process the simulation's outputs via the provided scripts:

- R and the rgdal library (shapefiles' data generation);
- Python and the mapnik2 and cairo libraries (shapefiles process);

- ffmpeg (animated maps generation);
- doxygen (documentation generation);
- gdb (the GNU debugger).

Note that some outputs of VirtualBelgium are compatible with MATSim[(1)] (Meister et al., 2010), an agent-based framework for transport simulation.

A.4 Compilation and execution

Workstation

To compile VirtualBelgium on a regular workstation, execute

make

invoking the Makefile's default rule. Once the compilation is done, then launching the simulation is invoked by the script

./run.sh NP

where NP is the number of desired parallel processes.

High Performance Computers - SLURM launch script

Depending on the cluster setup, the Makefile's default rule may have to be tuned. For instance, we provide an option designed specifically for the *Lemaitre 2* cluster of the *Consortium des Équipements de Calcul Intensif* (`http://www.ceci-hpc.be/`). Invoking

make ucl

will compile VirtualBelgium on this particular architecture.

The script *run_ lemaitre2.sh* can be used to submit an MPI parallel job on a cluster operating the SLURM management system by executing the

sbatch run_lemaitre2.sh

command. Various options can be set within the script, the most important one being

- mail-user : the mail address used for job notification;
- time : the requested run-time ;
- ntask : the number of process requested for the job ;
- mem-per-cpu : the memory requirement for the job, per process.

The job may requires the loading of several modules; check with your cluster administrator.

[(1)] available at `http://www.matsim.org`

High Performance Computers - SGE launch script

Depending on the cluster setup, the Makefile's default rule may have to be tuned. For instance, we provide an option designed specifically for the *Hercules* cluster of the *Consortium des Équipements de Calcul Intensif* (`http://www.ceci-hpc.be/`). Invoking

make

will compile VirtualBelgium on this particular architecture.

The script *run_clust.sh* can be used to submit an MPI parallel job on a cluster operating the Sun Grid Engine management system by executing the

qsub run_clust.sh

command. Various options can be set within the script, the most important one being

- M : the mail address used for job notification;
- h_cpu : the requested run-time ;
- pe openmpi : the number of process requested for the job ;
- vf : the memory requirement for the job, per process.

The job may requires the loading of several modules; check with your cluster administrator.

Debugging

While implementing new models in VirtualBelgium, you may want to use the a debugger. Compiling the project with

make debug

includes in the compiled binaries the debugging symbols necessary for the GNU gdb debugger. A parallel debugging session with NP processes can then be started by running

mpiexec -n NP xterm -e gdb ./vbel

in the ./bin directory.

A.5 VirtualBelgium configuration - model.props

VirtualBelgium contains a collection of models that can be activated and deactivated in the configuration file

./bin/model.props

which is also responsible for setting the models parameters and inputs data. The file is divided into 3 sections described next.

Parameters

This section specifies the simulation and checkpointing parameters.

par.start	Starting year of the simulation.
par.end	Ending year of the simulation.
par.debug	Debugging parameter (y = activated, not activated otherwise).
par.act_home	Code identifying the *return to home activity*.

Data files

The parameters in this section give the path to the various input files.

Socio-demographic data

file.ind	Baseline synthetic population (individuals).
file.hh	Baseline synthetic population (households).
file.age_dis_men	Men age distribution by municipality.
file.age_dis_women	Women age distribution by municipality.
file.mortality	Death probability by age and gender.

Road network data

file.network	Road network.
file.node_ins	Road network's nodes with their respective municipality id.
file.ins_id_code	Codebook for municipality name × id × INS code × district name.
file.indicators	Indicators by INS code, used for activity localization.

Activity-chains data

file.act_cdb	Activities codebook.

file.act_distance Parameters of the mixture of Normal distributions for the log(*distance*) achieved to reach an activity localization conditionally to the activity type.

file.act_duration Parameters of the mixture of Normal distributions for the log(*duration*) of an activity conditionally to the activity type.

file.act_tdep_house Parameters of the mixture of Normal distributions for the log(*house departure time*) of an activity conditionally to the activity type.

file.act_distance_duration Parameters of the mixture of bivariate Normal distributions for log(*distance*) × log(*duration*) attributes of an activity conditionally to the activity chain size and activity type.

file.start_duration Parameters of the mixture of bivariate Normal distributions for log(*start time*)×log(*duration*) attributes of an activity conditionally to the activity type.

file.dist_x_dur_trip Parameters of the mixture of bivariate Normal distributions for log(*distance*) × log(*duration*) of a trip.

file.start_x_end_time Parameters of the mixture of bivariate Normal distributions for *starting time* × *ending time* attributes of an activity conditionally to the activity type.

Models selection

Turning on and off the models of VirtualBelgium (y = activated, not activated otherwise).

evo.age Aging process.

evo.death Agents can die.

evo.birth Reproduction model.

evo.activity Activity-chain model for travel demand forecasting.

A.6 Inputs formats for transport demand forecasting

In this section, we detail the formats of the required inputs for the

Synthetic population

Agents generation in VirtualBelgium requires 2 files:

- one detailing the households. One line of this file is of the form

  id ins mun type n_childs n_adults list_ind

- and one for the individuals descriptions. One line of this file is of the form

  id ins mun hhtype gender spstatus dip drvlic age_cl act_chain

whose respective fields are described below.

Households description

id	Numerical id of the household.
ins	INS code of the household's municipality.
mun	Household's municipality name.
type	Household type. Possible values are *IH* (isolated man) *IF* (isolated woman) *F* (family), *C* (couple), *M* (mono-parental father) and *W* (mono-parental mother).
n_childs	Number of child in the household (0 to 5).
n_adults	Number of adults in the household (mate not included, 0 to 3).
list_ind	Listing of the individuals' id belonging to the household.

Individuals description

id	Numerical id of the individual.
ins	INS code of the individual's municipality.

mun	Individual's municipality name.
hhtype	Household type. Possible values are I (isolated), C (couple), F (family), N (mono-parental).
gender	Gender. Possible values are H (male) or F (female).
spstatus	Socio-professional status. Possible values are I (inactive), A (active) or E (student).
dip	Education level. Possible values are O (none), P (primary school), S (secondary school) and U (higher education).
drvlic	Driving licence ownership. Possible values are O (no) and P (yes).
age_cl	Age class. Possible values are 0 (0-5), 1 (6-17), 2, (18-39), 3 (40-59) and 4 (60+).
act_chain	Sequence of base activities. See the file pointed by the parameter *file.act_cdb* in the *model.props* configuration file for the list of possible activities.

Distributions

The activity-based model used to derive the traffic demand relies on the mixture of several univariate and bivariate Normal distributions. We now detail the encoding format of such distributions.

Mixture of univariate Normal distributions

Such distributions are characterized by

- an id;
- k univariate Normal distributions $\mathcal{N}(\mu_i, \sigma_i)$ $(i = 1, \ldots, k)$ where μ_i and σ_i are respectively the mean and standard deviation of the i^{th} distribution;
- a set of weights such that the i^{th} distribution is associated with weight $p_i \in [0, 1]$ and $\sum_i^k p_i = 1$. These weights are also referred as the mixing proportions.
- an upper bound max;

and are encoded in the following format:

$$id\ ;\ \mu_1\ ;\ \ldots\ ;\ \mu_k\ ;\ \sigma_1\ ;\ \ldots\ ;\ \sigma_k\ ;\ \ p_1\ ;\ \ldots\ ;\ p_k\ ;\ max$$

Mixture of bivariate Normal distributions

Similarly to the univariate case, these distributions are characterized by

- an id;
- k bivariate Normal distributions $\mathcal{N}(\boldsymbol{\mu_i}, \Sigma_i)$ where

$$\boldsymbol{\mu_i} = (\mu_{i,1}, \mu_{i,2}) \quad \text{and} \quad \Sigma_i = \begin{pmatrix} \sigma_{i,11} & \sigma_{i,12} \\ \sigma_{i,21} & \sigma_{i,22} \end{pmatrix}.$$

  are respectively the mean vector and variance-covariance matrix of the i^{th} distribution;
- and a set of weights such that the i^{th} distribution is associated with weight $p_i \in [0, 1]$ and $\sum_i^k p_i = 1$. These weights are also referred as the mixing proportions.
- a vector of upper bound $max = (max_1, max_2)$

and are encoded in the following format:

$$id \ ; \ \mu_{1,1} \ ; \ \dots \ ; \ \mu_{k,1} \ ; \ \mu_{1,2} \ ; \ \dots \ ; \ \mu_{k,2} \ ; \ \sigma_{1,11} \ ; \ \sigma_{1,12} \ ; \ \sigma_{1,21} \ ; \ \sigma_{1,22} \ ; \ \dots \ ; \ \sigma_{k,11} \ ; \ \sigma_{k,12} \ ; \ \sigma_{k,21} \ ; \ \sigma_{k,22} \ ; \ \dots \ ; \ p_1 \ ; \ \dots \ ; \ p_k \ ; \ max_1 \ ; \ max_2$$

Network

The road network is encoded as a XML file compatible with the input format of MATSim. An example of such file is given in Listing A.1. The necessary data can be extracted from OpenStreetMap and converted into a network XML file by using MATSim libraries[(2)].

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE network SYSTEM
      "http://www.matsim.org/files/dtd/network_v1.dtd">

<network name="example">

    <nodes>
        <node id="0" x="505046.8125" y="137967.7969" />
        <node id="1" x="520580.9063" y="147882.7969" />
        ...
    </nodes>

    <links capperiod="01:00:00" effectivecellsize="7.5"
          effectivelanewidth="3.75">
        <link id="0" from="0" to="1" length="6243.0"
```

[(2)] http://www.matsim.org/docs/tutorials/8lessons/input/creating/network

```
                freespeed="27.77" capacity="4000.0"
                permlanes="2.0" oneway="1" modes="car" />
        <link id="1" from="1" to="0" length="6243.0"
                freespeed="27.77" capacity="4000.0"
                permlanes="2.0" oneway="1" modes="car" />
        ...
    </links>

</network>
```

Listing A.1 – Network encoding (XML).

Note that *length* is given in meters, *capacity* in vehicles per hour and free-flow speed (*freespeed*) in meters per second. Number of lanes of a link correspond to the *permlanes* attributes. Every link are assumed to be one-way (*oneway="1"*) and can be travelled with a car (*mode="car"*).

A.7 Outputs for travel demand forecasting

Agents' agenda - activity_chains.xml

Listing A.2 illustrates an agent's agenda produced by the activity based model. This generated XML output is compatible with MATSim. Each entry *act* details an activity (type, localization and ending time). The *leg mode* attribute indicates which mode the individual used to reach the next activity.

```
<person id="9993331">
    <plan selected="yes">
        <act type="m" x="415857.564773" y="596350.923224"
                    end_time="11:25:24"/>
        <leg mode="car"/>
        <act type="c" x="410815.268373" y="595582.660781"
                    end_time="13:50:8"/>
        <leg mode="car"/>
        <act type="m" x="415857.564773" y="596350.923224"
                    end_time="14:18:41"/>
        <leg mode="car"/>
        <act type="l" x="416014.612566" y="595111.030150"
                    end_time="15:46:33"/>
        <leg mode="car"/>
        <act type="m" x="415857.564773" y="596350.923224"/>
    </plan>
</person>
```

Listing A.2 – An agent's schedule (XML).

Activities and trips data - activity_stat

VirtualBelgium provides a complete listing of every activity associated with a trip in a plain text file encoded in the following format:

type_act dist_trip dur_trip dur_act start_act size_chain id_chain

whose fields are detailed below.

type	Numerical type of the activity. See the file pointed by the parameter *file.act_cdb* in the *model.props* configuration file for the list of possible activities.
dist_trip	Distance performed to reach the activity localization (meters).
dur_trip	Duration of the trip (seconds).
dur_act	Duration of the activity (seconds).
start_act	Starting time of the activity (seconds elapsed since midnight).
size_chain	Size of the activity chain to which the current activity belongs.
id_chain	Rank of the activity in its chain.

Starting activities - activity_mun_start_time

The number of starting activities by municipality for each hour of the day is saved in a CSV file of the form:

ADMUKEY	H0	H1	H2	...	H23
11001	434	314	299	...	521
		⋮			
93090	1002	1347	1120	...	873

where the *ADMUNKEY* column corresponds to the INS codes of the municipalities, and the *H0* to *H23* columns are the hours of the day. Each line of the file represents then the evolution of the number of starting activities in a given municipality over one day.

Ending activities - activity_mun_end_time

Similarly to the starting activities, the number of ending activities by municipality for each hour of the day is saved in a comma-separated values file in the same form as the one used for recording the number of starting activities.

Origin-destination matrix - origin_destination

The origin-destination matrix (giving the flows between municipalities) is recorded in a comma-separated file, in which the first column and the first line respectively gives the origin and the destination of the flows. Note that the municipalities are identified by their INS code. An example of such file is given below.

```
O/D   ; 11001 ; 11002 ; ... ; 93090
11001 ; 20348 ;  9925 ; ... ;    1
                 ⋮
93090 ;    1   ;   84   ; ... ; 11644
```

Origin-destination array - origin_destination_array

The origin-destination matrix is also saved in an array format of which an example is given hereunder.

```
Origin , Dest  , Trips
11001  , 11001 , 20348
11001  , 11002 ,  9925
                 ⋮
93090  , 93090 , 11644
```

This file is suited to the JFlowMap software used to illustrate graphically and interactively the flows between municipalities (see Section A.8).

A.8 Post-processing scripts

VirtualBelgium provides various scripts to post-process the raw outputs of the simulation.

merge.sh	Shell script merging the activity-based model outputs created by the parallel processes into a single one.
merge_shp.R	R script adding the municipalities-related outputs from the activity-based model to a shapefile (provided in the ./data/network directory).
maps_generation.py	Python script that generates 24 maps illustrating the number of starting activities by municipality for each hour of a day. The maps are then merged into a movie representing the evolution over one day. Requires the data created by merge_shp.R.

od_one_municipality.pl Perl script that extracts the flows originating and going to one municipality from the origin-destination array created by the activity-based model. Takes one argument: the municipality INS code.

od_maps.sh Shell script launching the JFlowMap visualization software[3] from Boyandin et al. (2010). It represents graphically and interactively the flows between origins and destinations. Takes one argument: the name of an origin-destination array in the ./output directory. Figure A.1 illustrates a snapshot of the software.

A.9 Source documentation

Source documentation is either available on-line at

`http://virtualbelgium.sourceforge.net/doc/`

or in the files

./doc/html/index.html and ./doc/refman.pdf

A.10 SVN repository

The latest development version of the project is also available on a SVN repository hosted at the University of Namur (Belgium). Providing that you have a SVN client installed on your workstation, just follow these steps to access the repository :

1. request an user account at `grt@math.unamur.be`;
2. create an SSH tunnel to the Gauss server:

   ssh -N -f -l 5555:localhost:3690 user@gauss.math.fundp.ac.be

   and execute

   svn co svn://user@localhost:5555/var/svn/virtualbelgium/

   to retrieve latest development version of VirtualBelgium.
3. if the previous step does not work then execute:

   svn co svn+ssh://user@gauss.math.fundp.ac.be/var/svn/virtualbelgium --username user

[3] available at https://code.google.com/p/jflowmap/

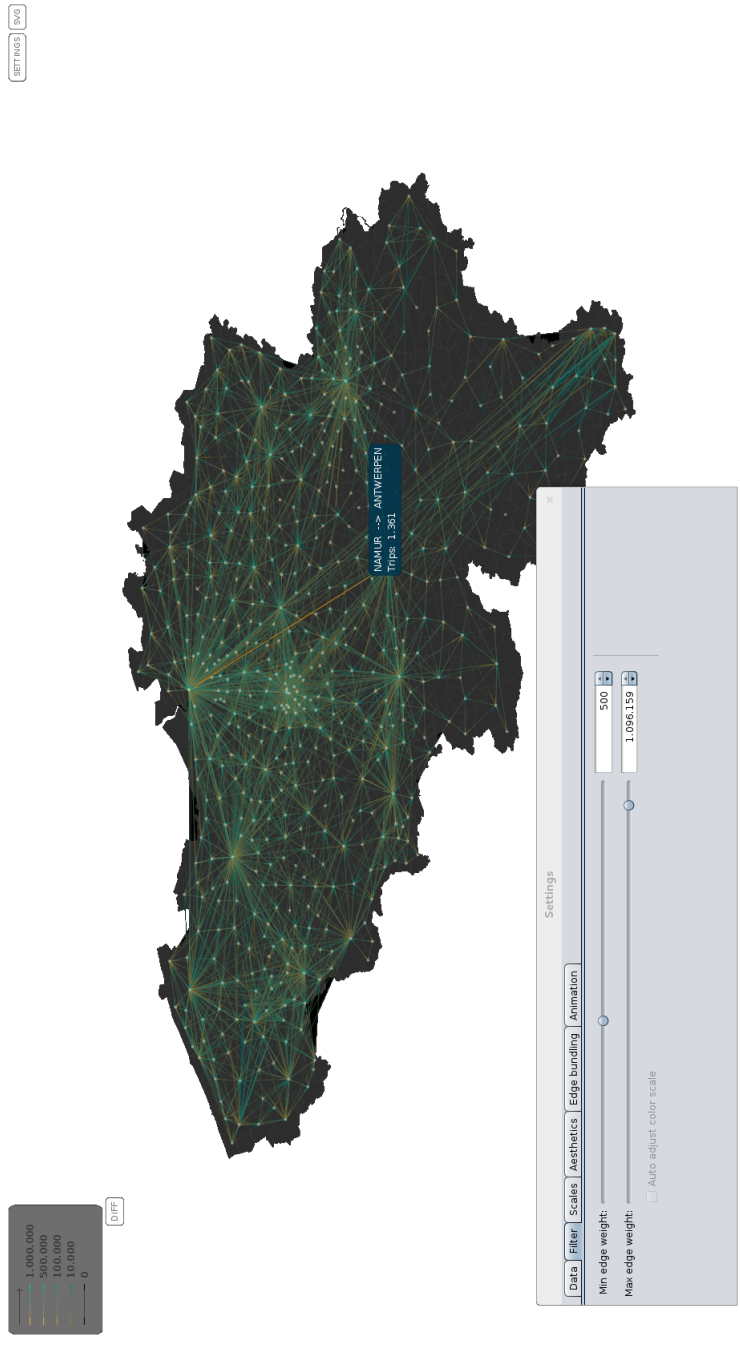

Figure A.1 – Screenshot of JFlowMap, an interactive tool for representing graphically origin-destination flows.

Appendix B

Synthetic population generator: technical details

B.1 Data sources

The synthetic population generator presented in Chapter 2 requires data which is provided from the following sources:

- *Directorate-general Statistics and Economic information* of the Belgian Federal Government (INS);
- *Service public fédéral Mobilité et Transports* of the Belgian Federal Government (DIV);
- *Groupe d'étude de démographie appliquée* (GéDAP) centre of the University of Louvain-la-Neuve (Belgium);
- the *MOBEL* mobility survey (Hubert and Toint, 2002).

We list them in the following.

Municipality characteristics

- Municipality name
- Municipality identifier (a number from 1 to 589)
- Municipality INS code
- District name
- District INS code
- Population (GéDAP, 2001):

  - total number of men
  - total number of women
  - total number of individuals

- Population by gender and age class (GéDAP, 2001)
- Driving licence (DIV, 2004): total number of individuals having a driving licence by gender and age class
- Population by household type (GéDAP, 2001)
- Population by activity status
- Population by diploma (GéDAP, 2001)
- Number of households by household type and household size (INS, 2001)
- Number of households by household type and number of children (INS, 2001)

District characteristics

- District name
- District INS code
- Province name
- Population by household type and age (INS, 2001)
- Population by diploma and age (INS, 2001)
- Population by activity status and age (INS, 2001)
- Population of 6+ years old individuals by activity status and diploma (INS, 2001)
- Population of 6+ years old individuals by gender and diploma (INS, 2001)
- Population by sex and activity (INS, 2001)

National characteristics - MOBEL

- Distribution of household's head's gender by age and household type
- Distribution of the age class for the household's head by municipality type and household type
- Distribution of the activity status for the household's head by municipality type, age class and household type

- Distribution of the household's head diploma by municipality type, age class and household type
- Distribution of the mate's age class as a function of the head's gender and age class by household type
- Distribution of the mate's diploma as a function of the head's gender and diploma by household type

B.2 SIF file for LANCELOT

The synthetic population generator needs to solve an optimization problem for each municipality (Section 2.3.2). The solution is determined using an augmented Lagrangian algorithm, as implemented in the (freely available) LANCELOT package (Conn et al., 1992 and Gould et al., 2003). The problem formulation in the LANCELOT syntax is given hereunder (SIF file).

```
***************************
* SET UP THE INITIAL DATA *
***************************

NAME          =PNAME

*   Problem :
*   *********

*   This problem arises in the determination of a synthetic
*   population for Belgian municipalities. It estimates, for
*   a given municipality, the number of households of the
*   following types:
*     type F (a couple)
*     type W (a woman)
*     type M (a man)
*   which have a specific number of children (1 to 5) and
*   additional adults (0 to 2).
*
*   The data consists in
*      - the number of individuals in households with 3
*        to 8 members,
*      - the number of F, W and N households according
*        to their number of children
*      - and the total number of individuals in F- and
*        W/M-households.
*
*   The variables are:
```

```
*
*   NF10: number of F with 1 child    and 0 additional adult,
*   NF11: number of F with 1 child    and 1 additional adult,
*   NF12: number of F with 1 child    and 2 additional adults,
*   NF20: number of F with 2 children and 0 additional adult,
*   NF21: number of F with 2 children and 1 additional adult,
*   NF32: number of F with 2 children and 2 additional adults,
*   NF30: number of F with 3 children and 0 additional adult,
*   NF31: number of F with 3 children and 1 additional adult,
*   NF32: number of F with 3 children and 2 additional adults,
*   NF40: number of F with 4 children and 0 additional adult,
*   NF41: number of F with 4 children and 1 additional adult,
*   NF42: number of F with 4 children and 2 additional adults,
*   NF50: number of F with 5 children and 0 additional adult,
*   NF51: number of F with 5 children and 1 additional adult,
*   NF52: number of F with 5 children and 2 additional adults,
*   NW10: number of W with 1 child    and 0 additional adult,
*   NW11: number of W with 1 child    and 1 additional adult,
*   NW12: number of W with 1 child    and 2 additional adults,
*   NW20: number of W with 2 children and 0 additional adult,
*   NW21: number of W with 2 children and 1 additional adult,
*   NW22: number of W with 2 children and 2 additional adults,
*   NW30: number of W with 3 children and 0 additional adult,
*   NW31: number of W with 3 children and 1 additional adult,
*   NW32: number of W with 3 children and 2 additional adults,
*   NM10: number of M with 1 child    and 0 additional adult,
*   NM11: number of M with 1 child    and 1 additional adult,
*   NM12: number of M with 1 child    and 2 additional adults,
*   NM20: number of M with 2 children and 0 additional adult,
*   NM21: number of M with 2 children and 1 additional adult,
*   NM32: number of M with 2 children and 2 additional adults,
*   NM30: number of M with 3 children and 0 additional adult,
*   NM31: number of M with 3 children and 1 additional adult,
*   NM32: number of M with 3 children and 2 additional adults,
*
*   all these variables being non-negative.
*   The other constraints are:
*
*   1) the number of households of size 3:
*
*      NF10 + NW20 + NW11 + NM20 + NM11 = M3
*
*   2) the number of households of size 4:
*
*      NF20 + NF11 + NW30 + NW21 + NW12 + NM30 + NM21 + NM12 = M4
```

```
*
*   3) the number of households of size 5:
*
*      NF30 + NF21 + NF12 + NW31 + NW22 + NM31 + NM22 = M5
*
*   4) the number of households of size 6:
*
*      NF40 + NF31 + NF22 + NW32 + NM32 = M6
*
*   5) the number of households of size 7:
*
*      NF50 + NF41 + NF32 = M7
*
*   6) the number of households of size 8 and more:
*
*      NF51 + NF42 + NF52 = M8
*
*   7) the number of F-households with 1 child:
*
*      NF10 + NF11 + NF12 = M1F
*
*   8) the number of F-households with 2 children:
*
*      NF20 + NF21 + NF22 = M2F
*
*   9) the number of F-households with 3 children or more:
*
*      NF30 + NF31 + NF32 + NF40 + NF41 + NF42 + NF50
*           + NF51 + NF52 = M3F
*
*   10) the number of W-households with 1 child:
*
*      NW10 + NW11 + NW12 = M1W
*
*   11) the number of W-households with 2 children:
*
*      NW20 + NW21 + NW22 = M2W
*
*   12) the number of W-households with 3 children or more:
*
*      NW30 + NW31 + NW32 = M3W
*
*   13) the number of M-households with 1 child:
*
*      NM10 + NM11 + NM12 = M1M
```

```
*
*   14) the number of M-households with 2 children:
*
*      NM20 + NM21 + NM22 = M2M
*
*   15) the number of M-households with 3 children or more:
*
*      NM30 + NM31 + NM32 = M3M
*
*   16) the number of individuals in F-households:
*
*      3*NF10 + 4*NF11 + 5*NF12 + 4*NF20 + 5*NF21 + 6*NF22
*             + 5*NF30 + 6*NF31 + 7*NF32 + 6*NF40 + 7*NF41
*             + 8*NF42 + 7*NF50 + 8*NF51 + 9*NF52 = NINF
*
*   17 ) the number of individuals in N-households:
*
*      2*NW10 + 3*NW11 + 4*NW12 + 3*NW20 + 4*NW21 + 5*NW22
*             + 4*NW30 + 5*NW31 + 6*NW32 + 2*NM10 + 3*NM11
*             + 4*NM12 + 3*NM20 + 4*NM21 + 5*NM22 + 4*NM30
*             + 5*NM31 + 6*NM32 = NINN
*
*   Equations are rescaled to that their right-hand side
*   is of the order of unity.
*

*   Problem initial data

*   Number of households according to their sizes

 RE M3                    ===M3===
 RE M4                    ===M4===
 RE M5                    ===M5===
 RE M6                    ===M6===
 RE M7                    ===M7===
 RE M8                    ===M8===

*  Number of F-households according the number of children

 RE M1F                   ===M1F==
 RE M2F                   ===M2F==
 RE M3F                   ===M3F==

*  Number of M-households according the number of children
```

```
 RE M1M                  ===M1M==
 RE M2M                  ===M2M==
 RE M3M                  ===M3M==

*  Number of W-households according the number of children


 RE M1W                  ===M1W==
 RE M2W                  ===M2W==
 RE M3W                  ===M3W==

*  Number of children and adults in households of type F and N

 RE NINF                 ===NINF=

*  Number of individuals in N-households

 RE NINN                 ===NINN=

*  Avoid zero right-hand sides, as their inverse is needed for
*  scaling and regularization.

 RA M3P1      M3         1.0
 RA M4P1      M4         1.0
 RA M5P1      M5         1.0
 RA M6P1      M6         1.0
 RA M7P1      M7         1.0
 RA M8P1      M8         1.0
 RA M1FP1     M1F        1.0
 RA M2FP1     M2F        1.0
 RA M3FP1     M3F        1.0
 RA M1WP1     M1W        1.0
 RA M2WP1     M2W        1.0
 RA M3WP1     M3W        1.0
 RA M1MP1     M1M        1.0
 RA M2MP1     M2M        1.0
 RA M3MP1     M3M        1.0

* The regularization parameter

 RE PENPAR               ===PP===

* Update the scalings for the penalty parameter

 R/ PM3       M3P1                    PENPAR
```

```
 R/ PM4       M4P1                    PENPAR
 R/ PM5       M5P1                    PENPAR
 R/ PM6       M6P1                    PENPAR
 R/ PM7       M7P1                    PENPAR
 R/ PM8       M8P1                    PENPAR
 R/ PM1F      M1FP1                   PENPAR
 R/ PM2F      M2FP1                   PENPAR
 R/ PM3F      M3FP1                   PENPAR
 R/ PM1W      M1WP1                   PENPAR
 R/ PM2W      M2WP1                   PENPAR
 R/ PM3W      M3WP1                   PENPAR
 R/ PM1M      M1MP1                   PENPAR
 R/ PM2M      M2MP1                   PENPAR
 R/ PM3M      M3MP1                   PENPAR

VARIABLES

    NF10
    NF11
    NF12
    NF20
    NF21
    NF22
    NF30
    NF31
    NF32
    NF40
    NF41
    NF42
    NF50
    NF51
    NF52
    NW10
    NW11
    NW12
    NW20
    NW21
    NW22
    NW30
    NW31
    NW32
    NM10
    NM11
    NM12
    NM20
```

```
    NM21
    NM22
    NM30
    NM31
    NM32

GROUPS

*   Household of size 3 to 8: each of these constraints
*   imposes that the total number of  households
*   of a given size is equal to the observed value.

 N  HSZ3      NF10      1.0            NW20      1.0
 N  HSZ3      NW11      1.0            NM20      1.0
 N  HSZ3      NM11      1.0
 ZN HSZ3      'SCALE'                  PM3

 N  HSZ4      NF20      1.0            NF11      1.0
 N  HSZ4      NW30      1.0            NW21      1.0
 N  HSZ4      NW12      1.0            NM30      1.0
 N  HSZ4      NM21      1.0            NM12      1.0
 ZN HSZ4      'SCALE'                  PM4

 N  HSZ5      NF30      1.0            NF21      1.0
 N  HSZ5      NF12      1.0            NW31      1.0
 N  HSZ5      NW22      1.0            NM31      1.0
 N  HSZ5      NM22      1.0
 ZN HSZ5      'SCALE'                  PM5

 N  HSZ6      NF40      1.0            NF31      1.0
 N  HSZ6      NF22      1.0            NW32      1.0
 N  HSZ6      NM32      1.0
 ZN HSZ6      'SCALE'                  PM6

 N  HSZ7      NF50      1.0            NF41      1.0
 N  HSZ7      NF32      1.0
 ZN HSZ7      'SCALE'                  PM7

 N  HSZ8      NF51      1.0            NF42      1.0
 N  HSZ8      NF52      1.0
 ZN HSZ8      'SCALE'                  PM8


*   Household with 1,2 or 3+ children, for families (F),
*   isolated woman (W) or isolated man (M)
```

```
 N  HST1F     NF10      1.0            NF11      1.0
 N  HST1F     NF12      1.0
 ZN HST1F     'SCALE'                  PM1F

 N  HST2F     NF20      1.0            NF21      1.0
 N  HST2F     NF22      1.0
 ZN HST2F     'SCALE'                  PM2F

 N  HST3F     NF30      1.0            NF31      1.0
 N  HST3F     NF32      1.0            NF40      1.0
 N  HST3F     NF41      1.0            NF42      1.0
 N  HST3F     NF50      1.0            NF51      1.0
 N  HST3F     NF52      1.0
 ZN HST3F     'SCALE'                  PM3F

 N  HST1W     NW10      1.0            NW11      1.0
 N  HST1W     NW12      1.0
 ZN HST1W     'SCALE'                  PM1W

 N  HST2W     NW20      1.0            NW21      1.0
 N  HST2W     NW22      1.0
 ZN HST2F     'SCALE'                  PM2W

 N  HST3W     NW30      1.0            NW31      1.0
 N  HST3W     NW32      1.0
 ZN HST3W     'SCALE'                  PM3W

 N  HST1M     NM10      1.0            NM11      1.0
 N  HST1M     NM12      1.0
 ZN HST1M     'SCALE'                  PM1M

 N  HST2M     NM20      1.0            NM21      1.0
 N  HST2M     NM22      1.0
 ZN HST2M     'SCALE'                  PM2M

 N  HST3M     NM30      1.0            NM31      1.0
 N  HST3M     NM32      1.0
 ZN HST3M     'SCALE'                  PM3M


*   Number of individuals in F-households

 E  HINF      NF10      3.0            NF11      4.0
 E  HINF      NF12      5.0            NF20      4.0
```

```
 E  HINF      NF21      5.0            NF22      6.0
 E  HINF      NF30      5.0            NF31      6.0
 E  HINF      NF32      7.0            NF40      6.0
 E  HINF      NF41      7.0            NF42      8.0
 E  HINF      NF50      7.0            NF51      8.0
 E  HINF      NF52      9.0
 E  HINF      'SCALE'   0.001

*   Number of individuals in N-households

 E  HINN      NW10      2.0            NW11      3.0
 E  HINN      NW12      4.0            NW20      3.0
 E  HINN      NW21      4.0            NW22      5.0
 E  HINN      NW30      4.0            NW31      5.0
 E  HINN      NW32      6.0            NM10      2.0
 E  HINN      NM11      3.0            NM12      4.0
 E  HINN      NM20      3.0            NM21      4.0
 E  HINN      NM22      5.0            NM30      4.0
 E  HINN      NM31      5.0            NM32      6.0
 E  HINN      'SCALE'   0.001

*   Linear terms of the entropy function

 N  GNF10     NF10      -1.0
 N  GNF11     NF11      -1.0
 N  GNF12     NF12      -1.0
 N  GNF20     NF20      -1.0
 N  GNF21     NF21      -1.0
 N  GNF22     NF22      -1.0
 N  GNF30     NF30      -1.0
 N  GNF31     NF31      -1.0
 N  GNF32     NF32      -1.0
 N  GNF40     NF40      -1.0
 N  GNF41     NF41      -1.0
 N  GNF42     NF42      -1.0
 N  GNF50     NF50      -1.0
 N  GNF51     NF51      -1.0
 N  GNF52     NF52      -1.0
 N  GNW10     NW10      -1.0
 N  GNW11     NW11      -1.0
 N  GNW12     NW12      -1.0
 N  GNW20     NW20      -1.0
 N  GNW21     NW21      -1.0
 N  GNW22     NW22      -1.0
 N  GNW30     NW30      -1.0
```

```
 N  GNW31     NW31      -1.0
 N  GNW32     NW32      -1.0
 N  GNM10     NM10      -1.0
 N  GNM11     NM11      -1.0
 N  GNM12     NM12      -1.0
 N  GNM20     NM20      -1.0
 N  GNM21     NM21      -1.0
 N  GNM22     NM22      -1.0
 N  GNM30     NM30      -1.0
 N  GNM31     NM31      -1.0
 N  GNM32     NM32      -1.0

CONSTANTS

 Z  =PNAME    HSZ3                    M3
 Z  =PNAME    HSZ4                    M4
 Z  =PNAME    HSZ5                    M5
 Z  =PNAME    HSZ6                    M6
 Z  =PNAME    HSZ7                    M7
 Z  =PNAME    HSZ8                    M8

 Z  =PNAME    HST1F                   M1F
 Z  =PNAME    HST2F                   M2F
 Z  =PNAME    HST3F                   M3F

 Z  =PNAME    HST1W                   M1W
 Z  =PNAME    HST2W                   M2W
 Z  =PNAME    HST3W                   M3W

 Z  =PNAME    HST1M                   M1M
 Z  =PNAME    HST2M                   M2M
 Z  =PNAME    HST3M                   M3M

 Z  =PNAME    HINF                    NINF
 Z  =PNAME    HINN                    NINN

BOUNDS

 XL =PNAME    'DEFAULT' 0.01

START POINT

 RM M1F/3     M1F       0.333333333
 XV =PNAME    NF10                    =M1F/3_1
 XV =PNAME    NF11                    =M1F/3_2
```

```
 XV =PNAME    NF12                    =M1F/3_3
 RM M2F/3     M2F       0.333333333
 XV =PNAME    NF20                    =M2F/3_1
 XV =PNAME    NF21                    =M2F/3_2
 XV =PNAME    NF22                    =M2F/3_3
 RM M3F/9     M3F       0.111111111
 XV =PNAME    NF30                    =M3F/9_1
 XV =PNAME    NF31                    =M3F/9_2
 XV =PNAME    NF32                    =M3F/9_3
 XV =PNAME    NF40                    =M3F/9_4
 XV =PNAME    NF41                    =M3F/9_5
 XV =PNAME    NF42                    =M3F/9_6
 XV =PNAME    NF50                    =M3F/9_7
 XV =PNAME    NF51                    =M3F/9_8
 XV =PNAME    NF52                    =M3F/9_9
 RM M1W/3     M1W       0.333333333
 XV =PNAME    NW10                    =M1W/3_1
 XV =PNAME    NW11                    =M1W/3_2
 XV =PNAME    NW12                    =M1W/3_3
 RM M2W/3     M2W       0.333333333
 XV =PNAME    NW20                    =M2W/3_1
 XV =PNAME    NW21                    =M2W/3_2
 XV =PNAME    NW22                    =M2W/3_3
 RM M3W/3     M3W       0.333333333
 XV =PNAME    NW30                    =M3W/3_1
 XV =PNAME    NW31                    =M3W/3_2
 XV =PNAME    NW32                    =M3W/3_3
 RM M1M/3     M1M       0.333333333
 XV =PNAME    NM10                    =M1M/3_1
 XV =PNAME    NM11                    =M1M/3_2
 XV =PNAME    NM12                    =M1M/3_3
 RM M2M/3     M2M       0.333333333
 XV =PNAME    NM20                    =M2M/3_1
 XV =PNAME    NM21                    =M2M/3_2
 XV =PNAME    NM22                    =M2M/3_3
 RM M3M/3     M3M       0.333333333
 XV =PNAME    NM30                    =M3M/3_1
 XV =PNAME    NM31                    =M3M/3_2
 XV =PNAME    NM32                    =M3M/3_3

ELEMENT TYPE

 EV ENTROP    X

ELEMENT USES
```

```
T  ENF10     ENTROP
V  ENF10     X                    NF10
T  ENF11     ENTROP
V  ENF11     X                    NF11
T  ENF12     ENTROP
V  ENF12     X                    NF12
T  ENF20     ENTROP
V  ENF20     X                    NF20
T  ENF21     ENTROP
V  ENF21     X                    NF21
T  ENF22     ENTROP
V  ENF22     X                    NF22
T  ENF30     ENTROP
V  ENF30     X                    NF30
T  ENF31     ENTROP
V  ENF31     X                    NF31
T  ENF32     ENTROP
V  ENF32     X                    NF32
T  ENF40     ENTROP
V  ENF40     X                    NF40
T  ENF41     ENTROP
V  ENF41     X                    NF41
T  ENF42     ENTROP
V  ENF42     X                    NF42
T  ENF50     ENTROP
V  ENF50     X                    NF50
T  ENF51     ENTROP
V  ENF51     X                    NF51
T  ENF52     ENTROP
V  ENF52     X                    NF52
T  ENW10     ENTROP
V  ENW10     X                    NW10
T  ENW11     ENTROP
V  ENW11     X                    NW11
T  ENW12     ENTROP
V  ENW12     X                    NW12
T  ENW20     ENTROP
V  ENW20     X                    NW20
T  ENW21     ENTROP
V  ENW21     X                    NW21
T  ENW22     ENTROP
V  ENW22     X                    NW22
T  ENW30     ENTROP
V  ENW30     X                    NW30
```

```
 T  ENW31     ENTROP
 V  ENW31     X                      NW31
 T  ENW32     ENTROP
 V  ENW32     X                      NW32
 T  ENM10     ENTROP
 V  ENM10     X                      NM10
 T  ENM11     ENTROP
 V  ENM11     X                      NM11
 T  ENM12     ENTROP
 V  ENM12     X                      NM12
 T  ENM20     ENTROP
 V  ENM20     X                      NM20
 T  ENM21     ENTROP
 V  ENM21     X                      NM21
 T  ENM22     ENTROP
 V  ENM22     X                      NM22
 T  ENM30     ENTROP
 V  ENM30     X                      NM30
 T  ENM31     ENTROP
 V  ENM31     X                      NM31
 T  ENM32     ENTROP
 V  ENM32     X                      NM32

GROUP TYPE

 GV L2        GVAR

GROUP USES

*   The original groups

 T  HSZ3      L2
 T  HSZ4      L2
 T  HSZ5      L2
 T  HSZ6      L2
 T  HSZ7      L2
 T  HSZ8      L2
 T  HST1F     L2
 T  HST2F     L2
 T  HST3F     L2
 T  HST1W     L2
 T  HST2W     L2
 T  HST3W     L2
 T  HST1M     L2
 T  HST2M     L2
```

```
 T  HST3M     L2

*   The entropy groups

 E  GNF10     ENF10
 E  GNF11     ENF11
 E  GNF12     ENF12
 E  GNF20     ENF20
 E  GNF21     ENF21
 E  GNF22     ENF22
 E  GNF30     ENF30
 E  GNF31     ENF31
 E  GNF32     ENF32
 E  GNF40     ENF40
 E  GNF41     ENF41
 E  GNF42     ENF42
 E  GNF50     ENF50
 E  GNF51     ENF51
 E  GNF52     ENF52
 E  GNW10     ENW10
 E  GNW11     ENW11
 E  GNW12     ENW12
 E  GNW20     ENW20
 E  GNW21     ENW21
 E  GNW22     ENW22
 E  GNW30     ENW30
 E  GNW31     ENW31
 E  GNW32     ENW32
 E  GNM10     ENM10
 E  GNM11     ENM11
 E  GNM12     ENM12
 E  GNM20     ENM20
 E  GNM21     ENM21
 E  GNM22     ENM22
 E  GNM30     ENM30
 E  GNM31     ENM31
 E  GNM32     ENM32

OBJECT BOUND

 LO =PNAME              0.0

*   Solution

*LO SOLTN               0.0
```

```
ENDATA

***********************
* SET UP THE FUNCTION *
* AND RANGE ROUTINES  *
***********************

ELEMENTS      =PNAME

INDIVIDUALS

 T  ENTROP
 F                      X * LOG( X )
 G  X                   1.0 + LOG( X )
 H  X          X        1.0 / X

ENDATA

*********************
* SET UP THE GROUPS *
* ROUTINE           *
*********************

GROUPS        =PNAME

INDIVIDUALS

 T  L2
 F                      GVAR * GVAR
 G                      GVAR + GVAR
 H                      2.0

ENDATA
```

Appendix C

Sioux-Falls characteristics

C.1 Road network

Origin	Destination	Capacity (veh/km)	Length (km)	Speed limit (km/h)
1	2	25900	6	60
1	3	23403	4	60
2	1	25900	6	60
2	6	4958	5	60
3	1	23403	4	60
3	4	17110	4	60
3	12	23403	4	60
4	3	17110	4	60
4	5	17782	2	60
4	11	4908	6	60
5	4	17782	2	60
5	6	4947	4	60
5	9	10000	5	60
6	2	4958	5	60
6	5	4947	4	60
6	8	4898	2	60
7	8	7841	3	60
7	18	23403	2	60
8	6	4898	2	60
8	7	7841	3	60
8	9	5050	10	60
8	16	5045	5	60
9	5	10000	5	60
9	8	5050	10	60
9	10	13915	3	60

Origin	**Destination**	**Capacity (veh/km)**	**Length (km)**	**Speed limit (km/h)**
10	9	13915	3	60
10	11	10000	5	60
10	15	13512	6	60
10	16	4854	4	60
10	17	4993	8	60
11	4	4908	6	60
11	10	10000	5	60
11	12	4908	6	60
11	14	4876	4	60
12	3	23403	4	60
12	11	4908	6	60
12	13	25900	3	60
13	12	25900	3	60
13	24	5091	4	60
14	11	4876	4	60
14	15	5127	5	60
14	23	4924	4	60
15	10	13512	6	60
15	14	5127	5	60
15	19	14564	3	60
15	22	9599	3	60
16	8	5045	5	60
16	10	4854	4	60
16	17	5229	2	60
16	18	19679	3	60
17	10	4993	8	60
17	16	5229	2	60
17	19	4823	2	60
18	7	23403	2	60
18	16	19679	3	60
18	20	23403	4	60
19	15	14564	3	60
19	17	4823	2	60
19	20	5002	4	60
20	18	23403	4	60
20	19	5002	4	60
20	21	5059	6	60
20	22	5075	5	60

Origin	Destination	Capacity (veh/km)	Length (km)	Speed limit (km/h)
21	20	5059	6	60
21	22	5229	2	60
21	24	4885	3	60
22	15	9599	3	60
22	20	5075	5	60
22	21	5229	2	60
22	23	5000	4	60
23	14	4924	4	60
23	22	5000	4	60
23	24	5078	2	60
24	13	5091	4	60
24	21	4885	3	60
24	23	5078	2	60

C.2 Origin-destination matrix

```
<TOTAL OD FLOW> 360600.0

Origin  1
    1 :    0.0;  2 :  100.0;  3 :  100.0;  4 :  500.0;  5 :  200.0;
    6 :  300.0;  7 :  500.0;  8 :  800.0;  9 :  500.0; 10 : 1300.0;
   11 :  500.0; 12 :  200.0; 13 :  500.0; 14 :  300.0; 15 :  500.0;
   16 :  500.0; 17 :  400.0; 18 :  100.0; 19 :  300.0; 20 :  300.0;
   21 :  100.0; 22 :  400.0; 23 :  300.0; 24 :  100.0;

Origin  2
    1 :  100.0;  2 :    0.0;  3 :  100.0;  4 :  200.0;  5 :  100.0;
    6 :  400.0;  7 :  200.0;  8 :  400.0;  9 :  200.0; 10 :  600.0;
   11 :  200.0; 12 :  100.0; 13 :  300.0; 14 :  100.0; 15 :  100.0;
   16 :  400.0; 17 :  200.0; 18 :    0.0; 19 :  100.0; 20 :  100.0;
   21 :    0.0; 22 :  100.0; 23 :    0.0; 24 :    0.0;

Origin  3
    1 :  100.0;  2 :  100.0;  3 :    0.0;  4 :  200.0;  5 :  100.0;
    6 :  300.0;  7 :  100.0;  8 :  200.0;  9 :  100.0; 10 :  300.0;
   11 :  300.0; 12 :  200.0; 13 :  100.0; 14 :  100.0; 15 :  100.0;
   16 :  200.0; 17 :  100.0; 18 :    0.0; 19 :    0.0; 20 :    0.0;
   21 :    0.0; 22 :  100.0; 23 :  100.0; 24 :    0.0;

Origin  4
    1 :  500.0;  2 :  200.0;  3 :  200.0;  4 :    0.0;  5 :  500.0;
    6 :  400.0;  7 :  400.0;  8 :  700.0;  9 :  700.0; 10 : 1200.0;
   11 : 1400.0; 12 :  600.0; 13 :  600.0; 14 :  500.0; 15 :  500.0;
```

```
   16 :  800.0; 17 :  500.0; 18 :  100.0; 19 :  200.0; 20 :  300.0;
   21 :  200.0; 22 :  400.0; 23 :  500.0; 24 :  200.0;

Origin  5
    1 :  200.0;  2 :  100.0;  3 :  100.0;  4 :  500.0;  5 :    0.0;
    6 :  200.0;  7 :  200.0;  8 :  500.0;  9 :  800.0; 10 : 1000.0;
   11 :  500.0; 12 :  200.0; 13 :  200.0; 14 :  100.0; 15 :  200.0;
   16 :  500.0; 17 :  200.0; 18 :    0.0; 19 :  100.0; 20 :  100.0;
   21 :  100.0; 22 :  200.0; 23 :  100.0; 24 :    0.0;

Origin  6
    1 :  300.0;  2 :  400.0;  3 :  300.0;  4 :  400.0;  5 :  200.0;
    6 :    0.0;  7 :  400.0;  8 :  800.0;  9 :  400.0; 10 :  800.0;
   11 :  400.0; 12 :  200.0; 13 :  200.0; 14 :  100.0; 15 :  200.0;
   16 :  900.0; 17 :  500.0; 18 :  100.0; 19 :  200.0; 20 :  300.0;
   21 :  100.0; 22 :  200.0; 23 :  100.0; 24 :  100.0;

Origin  7
    1 :  500.0;  2 :  200.0;  3 :  100.0;  4 :  400.0;  5 :  200.0;
    6 :  400.0;  7 :    0.0;  8 : 1000.0;  9 :  600.0; 10 : 1900.0;
   11 :  500.0; 12 :  700.0; 13 :  400.0; 14 :  200.0; 15 :  500.0;
   16 : 1400.0; 17 : 1000.0; 18 :  200.0; 19 :  400.0; 20 :  500.0;
   21 :  200.0; 22 :  500.0; 23 :  200.0; 24 :  100.0;

Origin  8
    1 :  800.0;  2 :  400.0;  3 :  200.0;  4 :  700.0;  5 :  500.0;
    6 :  800.0;  7 : 1000.0;  8 :    0.0;  9 :  800.0; 10 : 1600.0;
   11 :  800.0; 12 :  600.0; 13 :  600.0; 14 :  400.0; 15 :  600.0;
   16 : 2200.0; 17 : 1400.0; 18 :  300.0; 19 :  700.0; 20 :  900.0;
   21 :  400.0; 22 :  500.0; 23 :  300.0; 24 :  200.0;

Origin  9
    1 :  500.0;  2 :  200.0;  3 :  100.0;  4 :  700.0;  5 :  800.0;
    6 :  400.0;  7 :  600.0;  8 :  800.0;  9 :    0.0; 10 : 2800.0;
   11 : 1400.0; 12 :  600.0; 13 :  600.0; 14 :  600.0; 15 :  900.0;
   16 : 1400.0; 17 :  900.0; 18 :  200.0; 19 :  400.0; 20 :  600.0;
   21 :  300.0; 22 :  700.0; 23 :  500.0; 24 :  200.0;

Origin  10
    1 : 1300.0;  2 :  600.0;  3 :  300.0;  4 : 1200.0;  5 : 1000.0;
    6 :  800.0;  7 : 1900.0;  8 : 1600.0;  9 : 2800.0; 10 :    0.0;
   11 : 4000.0; 12 : 2000.0; 13 : 1900.0; 14 : 2100.0; 15 : 4000.0;
   16 : 4400.0; 17 : 3900.0; 18 :  700.0; 19 : 1800.0; 20 : 2500.0;
   21 : 1200.0; 22 : 2600.0; 23 : 1800.0; 24 :  800.0;

Origin  11
    1 :  500.0;  2 :  200.0;  3 :  300.0;  4 : 1500.0;  5 :  500.0;
    6 :  400.0;  7 :  500.0;  8 :  800.0;  9 : 1400.0; 10 : 3900.0;
   11 :    0.0; 12 : 1400.0; 13 : 1000.0; 14 : 1600.0; 15 : 1400.0;
```

```
   16 : 1400.0; 17 : 1000.0; 18 :  100.0; 19 :  400.0; 20 :  600.0;
   21 :  400.0; 22 : 1100.0; 23 : 1300.0; 24 :  600.0;

Origin  12
    1 :  200.0;  2 :  100.0;  3 :  200.0;  4 :  600.0;  5 :  200.0;
    6 :  200.0;  7 :  700.0;  8 :  600.0;  9 :  600.0; 10 : 2000.0;
   11 : 1400.0; 12 :    0.0; 13 : 1300.0; 14 :  700.0; 15 :  700.0;
   16 :  700.0; 17 :  600.0; 18 :  200.0; 19 :  300.0; 20 :  400.0;
   21 :  300.0; 22 :  700.0; 23 :  700.0; 24 :  500.0;

Origin  13
    1 :  500.0;  2 :  300.0;  3 :  100.0;  4 :  600.0;  5 :  200.0;
    6 :  200.0;  7 :  400.0;  8 :  600.0;  9 :  600.0; 10 : 1900.0;
   11 : 1000.0; 12 : 1300.0; 13 :    0.0; 14 :  600.0; 15 :  700.0;
   16 :  600.0; 17 :  500.0; 18 :  100.0; 19 :  300.0; 20 :  600.0;
   21 :  600.0; 22 : 1300.0; 23 :  800.0; 24 :  800.0;

Origin  14
    1 :  300.0;  2 :  100.0;  3 :  100.0;  4 :  500.0;  5 :  100.0;
    6 :  100.0;  7 :  200.0;  8 :  400.0;  9 :  600.0; 10 : 2100.0;
   11 : 1600.0; 12 :  700.0; 13 :  600.0; 14 :    0.0; 15 : 1300.0;
   16 :  700.0; 17 :  700.0; 18 :  100.0; 19 :  300.0; 20 :  500.0;
   21 :  400.0; 22 : 1200.0; 23 : 1100.0; 24 :  400.0;

Origin  15
    1 :  500.0;  2 :  100.0;  3 :  100.0;  4 :  500.0;  5 :  200.0;
    6 :  200.0;  7 :  500.0;  8 :  600.0;  9 : 1000.0; 10 : 4000.0;
   11 : 1400.0; 12 :  700.0; 13 :  700.0; 14 : 1300.0; 15 :    0.0;
   16 : 1200.0; 17 : 1500.0; 18 :  200.0; 19 :  800.0; 20 : 1100.0;
   21 :  800.0; 22 : 2600.0; 23 : 1000.0; 24 :  400.0;

Origin  16
    1 :  500.0;  2 :  400.0;  3 :  200.0;  4 :  800.0;  5 :  500.0;
    6 :  900.0;  7 : 1400.0;  8 : 2200.0;  9 : 1400.0; 10 : 4400.0;
   11 : 1400.0; 12 :  700.0; 13 :  600.0; 14 :  700.0; 15 : 1200.0;
   16 :    0.0; 17 : 2800.0; 18 :  500.0; 19 : 1300.0; 20 : 1600.0;
   21 :  600.0; 22 : 1200.0; 23 :  500.0; 24 :  300.0;

Origin  17
    1 :  400.0;  2 :  200.0;  3 :  100.0;  4 :  500.0;  5 :  200.0;
    6 :  500.0;  7 : 1000.0;  8 : 1400.0;  9 :  900.0; 10 : 3900.0;
   11 : 1000.0; 12 :  600.0; 13 :  500.0; 14 :  700.0; 15 : 1500.0;
   16 : 2800.0; 17 :    0.0; 18 :  600.0; 19 : 1700.0; 20 : 1700.0;
   21 :  600.0; 22 : 1700.0; 23 :  600.0; 24 :  300.0;

Origin  18
    1 :  100.0;  2 :    0.0;  3 :    0.0;  4 :  100.0;  5 :    0.0;
    6 :  100.0;  7 :  200.0;  8 :  300.0;  9 :  200.0; 10 :  700.0;
   11 :  200.0; 12 :  200.0; 13 :  100.0; 14 :  100.0; 15 :  200.0;
```

```
   16 :  500.0; 17 :  600.0; 18 :    0.0; 19 :  300.0; 20 :  400.0;
   21 :  100.0; 22 :  300.0; 23 :  100.0; 24 :    0.0;

Origin  19
    1 :  300.0;  2 :  100.0;  3 :    0.0;  4 :  200.0;  5 :  100.0;
    6 :  200.0;  7 :  400.0;  8 :  700.0;  9 :  400.0; 10 : 1800.0;
   11 :  400.0; 12 :  300.0; 13 :  300.0; 14 :  300.0; 15 :  800.0;
   16 : 1300.0; 17 : 1700.0; 18 :  300.0; 19 :    0.0; 20 : 1200.0;
   21 :  400.0; 22 : 1200.0; 23 :  300.0; 24 :  100.0;

Origin  20
    1 :  300.0;  2 :  100.0;  3 :    0.0;  4 :  300.0;  5 :  100.0;
    6 :  300.0;  7 :  500.0;  8 :  900.0;  9 :  600.0; 10 : 2500.0;
   11 :  600.0; 12 :  500.0; 13 :  600.0; 14 :  500.0; 15 : 1100.0;
   16 : 1600.0; 17 : 1700.0; 18 :  400.0; 19 : 1200.0; 20 :    0.0;
   21 : 1200.0; 22 : 2400.0; 23 :  700.0; 24 :  400.0;

Origin  21
    1 :  100.0;  2 :    0.0;  3 :    0.0;  4 :  200.0;  5 :  100.0;
    6 :  100.0;  7 :  200.0;  8 :  400.0;  9 :  300.0; 10 : 1200.0;
   11 :  400.0; 12 :  300.0; 13 :  600.0; 14 :  400.0; 15 :  800.0;
   16 :  600.0; 17 :  600.0; 18 :  100.0; 19 :  400.0; 20 : 1200.0;
   21 :    0.0; 22 : 1800.0; 23 :  700.0; 24 :  500.0;

Origin  22
    1 :  400.0;  2 :  100.0;  3 :  100.0;  4 :  400.0;  5 :  200.0;
    6 :  200.0;  7 :  500.0;  8 :  500.0;  9 :  700.0; 10 : 2600.0;
   11 : 1100.0; 12 :  700.0; 13 : 1300.0; 14 : 1200.0; 15 : 2600.0;
   16 : 1200.0; 17 : 1700.0; 18 :  300.0; 19 : 1200.0; 20 : 2400.0;
   21 : 1800.0; 22 :    0.0; 23 : 2100.0; 24 : 1100.0;

Origin  23
    1 :  300.0;  2 :    0.0;  3 :  100.0;  4 :  500.0;  5 :  100.0;
    6 :  100.0;  7 :  200.0;  8 :  300.0;  9 :  500.0; 10 : 1800.0;
   11 : 1300.0; 12 :  700.0; 13 :  800.0; 14 : 1100.0; 15 : 1000.0;
   16 :  500.0; 17 :  600.0; 18 :  100.0; 19 :  300.0; 20 :  700.0;
   21 :  700.0; 22 : 2100.0; 23 :    0.0; 24 :  700.0;

Origin  24
    1 :  100.0;  2 :    0.0;  3 :    0.0;  4 :  200.0;  5 :    0.0;
    6 :  100.0;  7 :  100.0;  8 :  200.0;  9 :  200.0; 10 :  800.0;
   11 :  600.0; 12 :  500.0; 13 :  700.0; 14 :  400.0; 15 :  400.0;
   16 :  300.0; 17 :  300.0; 18 :    0.0; 19 :  100.0; 20 :  400.0;
   21 :  500.0; 22 : 1100.0; 23 :  700.0; 24 :    0.0;
```

Bibliography

T. Adler and M. Ben-Akiva. A theoretical and empirical model of trip chaining behavior. *Transportation Research B*, **13**(3), 477–500, 1979.

T. A. Arentze and H. J. P. Timmermans. Albatross: A learning-based transportation oriented simulation system. Technical report, European Institute of Retailing and Services Studies. Eindhoven, The Netherlands, 2000.

T. Arentze, H. Timmermans, and F. Hofman. Creating synthetic household populations: Problems and approach. Paper presented at the 86th Transportation Research Board conference, Washington DC, US, 2007.

J. Auld, A. K. Mohammadian, and K. Wies. An efficient methodology for generating synthetic populations with multiple control levels. *Presented at the 89th Transportation Research Board Annual Meeting*, 2010.

L. Avery. *National Travel Survey: 2010.* National Travel Survey. Department for Transport, 2011.

M. Balmer, M. Rieser, K. Meister, D. Charypar, N. Lefebvre, K. Nagel, and K. Axhausen. Matsim-t: Architecture and simulation times. *Multi-agent systems for traffic and transportation engineering*, pp. 57–78, 2009.

H. Bar-Gera. Origin-based algorithm for the traffic assignment problem. *Transportation Science*, **36**(4), 398–417, 2002.

J. Barceló and J. Casas. Dynamic network simulation with aimsun. *in* 'Simulation Approaches in Transportation Analysis', pp. 57–98. Springer, 2005.

J. Barthelemy and Ph. L. Toint. Synthetic population generation without a sample. *Transportation Science*, **47**(2), 266–279, 2013.

A. Bazghandi. Techniques, advantages and problems of agent based modeling for traffic simulation. *International Journal of Computer Science*, **9**(3), 2012.

R. J. Beckman, K. A. Baggerly, and M. D. McKay. Creating synthetic baseline populations. *Transportation Research A*, **30**(6), 415–429, 1996.

S. Bekhor, C. Doblert, and K. W. Axhausen. *Integration of activity-based with agent-based models: an example from the Tel Aviv model and MATSim.* ETH Zürich, Institut für Verkehrsplanung, Transporttechnik, Strassen- und Eisenbahnbau (IVT), 2010.

M. Ben-Akiva, M. Bierlaire, H. Koutsopoulos, and R. Mishalani. Dynamit: a simulation-based system for traffic prediction. *in* 'DACCORS Short Term Forecasting Workshop, The Netherlands'. Citeseer, 1998.

R. F. Benekohal and J. Treiterer. Carsim: Car-following model for simulation of traffic in normal and stop-and-go conditions. *Transportation research record*, **1194**, 1988.

W. Benromach, J. Boreux, and J. Barthélemy. On the sublimation of whisky tasting. *Review of good old memories*, **42**(3), 14–15, 2014.

C. R. Bhat and F. S. Koppelman. Activity-based modeling of travel demand. *in* R. W. Hall, ed., 'Handbook of Transportation Science', pp. 35–61, Dordrecht, The Netherlands, 1999. Kluwer Academic Publishers.

C. R. Bhat, J. Y. Guo, A. Srinivasan, and A. Sivakumar. A Comprehensive Econometric Microsimulator for Daily Activity-Travel Patterns. *Transportation Research Record*, **1984**, 57–66, 2004.

M. Bierlaire. Evaluation de la demande en trafic : quelques méthodes de distribution. *Annales de la Société Scientifique de Bruxelles*, **105**(1-2), 17–66, 1991.

P. Bonsall. The influence of route guidance advice on route choice in urban networks. *Transportation*, **19**(1), 1–23, 1992.

P. Bonsall and A. D. May. Route choice in congested urban networks. *in* 'Research for Tomorrow's Transport Requirements. Proceedings of the Fourth World Conference on Transport Research', 1986.

I. Boyandin, E. Bertini, and D. Lalanne. Using flow maps to explore migrations over time. *in* 'Proceedings of Geospatial Visual Analytics Workshop in conjunction with The 13th AGILE International Conference on Geographic Information Science (GeoVA)', Guimaraes (Portugal), 2010.

M. Bradley, J. L. Bowman, and B. Griesenbeck. SACSIM: An applied activity-based model system with fine-level spatial and temporal resolution. *Journal of Choice Modelling*, **3**, 2010.

Bureau of Public Roads. *Traffic Assignment Manual.* U.S. Dept. of Commerce, Urban Planning Division, Washington D.C., 1964.

F. S. Chapin. *Human activity patterns in the city: Thing people do in time and space*, Vol. 13 of *Wiley series in urban research.* J. Wiley and Sons, 1974.

L. S. Chin, D. J. Worth, C. Greenough, S. Coakley, M. Holcombe, and M. Gheorghe. *FLAME-II: A Redesign of the Flexible Large-scale Agent-based Modelling Environment.* STFC, 2012.

Y. Chiu, J. Bottom, M. Mahut, A. Paz, R. Balakrishna, T. Waller, and J. Hicks. Dynamic traffic assignment: A primer. *Transportation Research E-Circular*, **E-C153**, 2011.

N. T. Collier and M. North. Parallel agent-based simulation with repast for high performance computing. *SIMULATION*, 2012.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. LANCELOT*: a Fortran package for large-scale nonlinear optimization (Release A).* Number 17 *in* 'Springer Series in Computational Mathematics'. Springer Verlag, Heidelberg, Berlin, New York, 1992.

E. Cornélis and Ph. L. Toint. Pacsim: a new dynamic behavioural model for multimodal traffic assignment. *in* M. Labbé, G. Laporte, K. Tanczos and P. L. Toint, eds, 'Operations Research and Decision Aid Methodologies in Traffic and Transportation Management', pp. 28–45, Heidelberg, Berlin, New York, 1998. Springer Verlag.

E. Cornelis, M. Hubert, Ph. Huynen, K. Lebrun, G. Patriarche, A. De Witte, L. Creemers, K. Declercq, D. Janseens, M. Castaigne, L. Hollaert, and F. Walle. La mobilité en belgique en 2010 : résultats de l'enquête beldam. Technical report, SPF Mobilité et Transports and BELSPO, Brussels, Belgium, 2012.

E. Cornélis, L. Legrain, and Ph. L. Toint. Synthetic populations: a tool for estimating travel demand. *in* B. Jourquin, ed., 'BIVEC-GIBET Transport Research Day 2005', Vol. 1, pp. 217–235. VUBPRESS Brussels University Press, 2005.

C. F. Daganzo and Y. Sheffi. On stochastic models of traffic assignment. *Transportation Science*, **11**(3), 253–274, 1977.

Ph. Dehoux and Ph. Toint. Some comments on dynamic modelling, in the presence of advanced driver information systems. *in* G. Argyrakos, M. Carrara, O. Cartsen, P. Davies, W. Mohlenbrink, M. Papageorgiou, T. Rothengatter and P. L. Toint, eds, 'Advanced Telematics in Road Transport', pp. 964–981. Commission of the European Communities - DG XIII, 1991.

W. E. Deming and F. F. Stephan. A least squares adjustment of a sampled frequency table when the expected marginal totals are known. *Annals of Mathematical Statistics*, **11**, 428–444, 1940.

J. W. Dickey. *Metropolitan Transportation Planning.* McGraw-Hill, New York, USA, 2 edn, 1983.

Direction Régionale de l'Équipement d'Île-de-France. *Les déplacement des Franciliens en 2001–2002, Enquê Globale de Transport.* Documentation Française, Paris, France, 2004.

T. Domencich and D. McFadden. *Urban Travel Demand: A Behavioural Analysis.* North Holland, Amsterdam, The Netherlands, 1975.

R. M. Downs and D. Stea. *Maps in minds.* Harper and Row, New York, 1977.

W. Dubitzky, K. Kurowski, and B. Schott. *Large-scale computing techniques for complex system simulations*, Vol. 80. Wiley. com, 2012.

H. B. Dwight. *Tables of integrals and other mathematical data.* The Macmillan Company, fourth edn, 1961.

M. L. Eaton. *Multivariate statistics: a vector space approach*, pp. 116–117. Wiley New York, 1983.

A. Eiben and J. E. Smith. *Introduction to Evolutionary Computing.* Springer, 2003.

G. Fasano and A. Franceschini. A multidimensional version of the Kolmogorov-Smirnov test. *Monthly Notices of the Royal Astronomical Society*, **225**, 155–170, 1987.

M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, **34**(3), 596–615, 1987.

M. Frick and K. Axhausen. Generating synthetic populations using IPF and Monte-Carlo techniques: some new results. 4th Swiss Transport Research Conference, Monte-Verita, 2004.

T. L. Friesz, D. Bernstein, T. E. Smith, R. L. Tobin, and B. W.Wie. A variational inequality formulation of the dynamic network user equilibrium problem. *Operations Research*, **41**(1), 179–191, 1993.

L. P. Gan and W. Recker. A mathematical programming formulation of the household activity rescheduling problem. *Transportation Research Part B: Methodological*, **42**(6), 571 – 606, 2008.

S. Gao, E. Frejinger, and M. Ben-Akiva. Adaptive route choices in risky traffic networks: A prospect theory approach. *Transportation research part C: emerging technologies*, **18**(5), 727–740, 2010*a*.

W. Gao, M. Balmer, and E. J. Miller. Comparison of matsim and emme/2 on greater toronto and hamilton area network, canada. *Transportation Research Record: Journal of the Transportation Research Board*, **2197**(1), 118–128, 2010*b*.

F. Gargiulo, S. Ternes, S. Huet, and G. Deffuant. An iterative approach for generating statistically realistic populations of households. *PLoS ONE*, **5**(1), 01 2010.

F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, **13**, 533–549, 1986.

F. Glover. Tabu search - Part I. *ORSA Journal on Computing*, **1**, 190–206, 1989.

F. Glover. Tabu search - Part II. *ORSA Journal on Computing*, **2**, 4–32, 1990.

F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, Boston, 1997.

T. F. Golob. Structural equation modeling for travel behavior research. *Transportation Research Part B: Methodological*, **37**(1), 1 – 25, 2003.

J. Goran. Activity based travel demand modelling - a literature study. Technical report, Danmarks TransportForskning, 2001.

N. I. M. Gould, D. Orban, and Ph. L. Toint. GALAHAD—a library of thread-safe Fortran 90 packages for large-scale nonlinear optimization. *ACM Transactions on Mathematical Software*, **29**(4), 353–372, 2003.

Y. Guo and C. R. Bhat. Population synthesis for microsimulating travel behavior. *Transportation Research Record: Journal of the Transportation Research Board*, **2014**, 92–101, 2007.

T. Hägerstrand. What about people in regional science. *Papers of the Regional Science*, **4**(1), 6–21, 1970.

M. M. Haklay and P. Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, **7**(4), 12–18, 2008.

S. L. Hoe. Issues and procedures in adopting structural equation modeling technique. *Journal of Applied quantitative methods*, **3**(1), 76–83, 2008.

A. Horni, D. M. Scott, M. Balmer, and K. W. Axhausen. Location choice modeling for shopping and leisure activities with matsim. *Transportation Research Record: Journal of the Transportation Research Board*, **2135**(1), 87–95, 2009.

Z. Huang and P. Williamson. A comparison of synthetic reconstruction and combinatorial optimization approaches to the creation of small-area microdata. Working Paper. Department of Geography, University of Liverpool, 2002.

J.-P. Hubert and Ph. L. Toint. *La mobilité quotidienne des Belges*. Number 1 *in* 'Mobilité et Transports'. Presses Universitaires de Namur, Namur, Belgium, 2002.

N. Huynh, M. Namazi-Rad, P. Perez, M. J. Berryman, Q. Chen, and J. Barthélemy. Generating a synthetic population in support of agent-based modeling of transportation in sydney. Technical report, SMART Infrastructure Facility, 2013.

C. T. Ireland and S. Kullback. Contingency tables with given marginals. *Biometrika*, **55**(1), 179–199, 1968.

R. Kitamura, C. Chen, and R. M. Pendyala. Generation of synthetic daily activity-travel patterns. *Transportation Research Record*, **1607**, 154–162, 1997.

R. Kitamura, E. Pas, C. Lula, T. K. Lawton, and P. Benson. The sequenced activity mobility simulator (SAMS): an integrated approac to modelling transportation, land use and air quality. *Transportation*, **23**, 267–291, 1996.

M. T. Koehler and B. F. Tivnan. Clustered computing with NetLogo and Repast J: Beyond chewing gum and duct tape. *in* 'Proc. of the Agents 2005 Conference on Generative Social Processes, Models, and Mechanisms', pp. 43–54, 2005.

E. Kreyszig. *Advanced engineering mathematics*. J. Wiley and Sons, Chichester, England, third edn, 1972.

D. Kriesel. *A Brief Introduction to Neural Networks*. on-line, 2007. available at http://www.dkriesel.com.

M. Lenormand and G. Deffuant. Generating a synthetic population of individuals in households: Sample-free vs sample-based methods. *Journal of Artificial Societies and Social Simulation*, **16**(4), 12, 2013.

P. S. Levy and S. Lemeshow. *Sampling of Populations - Methods and applications*. Wiley-Interscience Pubication, U.S.A., third edn, 1999.

R. J. A. Little and M.-M. Wu. Models for contingency tables with known margins when target and sampled population differ. *Journal of the American Statistical Association*, **86**(413), 87–95, 1991.

M. Lysenko and R. M. D'Souza. A framework for megascale agent based model simulations on graphics processing units. *Journal of Artificial Societies and Social Simulation*, **11**(4), 10, 2008.

F. Massaioli, F. Castiglione, and M. Bernaschi. Openmp parallelization of agent-based models. *Parallel Computing*, **31**(10), 1066–1081, 2005.

F. J. Massey. The kolmogorov-smirnov test for goodness of fit. *Journal of the American Statistical Association*, **46**(253), 68–78, 1951.

T. V. Mathew and K. V. Krishna Rao. Introduction to transportation engineering. *Civil Engineering–Transportation Engineering. IIT Bombay, NPTEL Online, http://www. cdeep. iitb. ac. in/nptel/Civil% 20Engineering*, 2007.

J. H. Mathewson, D. L. Trautman, and D. L. Gerlough. *Study of Traffic Flow by Simulation.* Reprint / Institute of Transportation and Traffic Engineering. Institute of Transportation and Traffic Engineering, University of California, 1955.

G. McLachlan and D. Peel. *Finite Mixture Models.* J. Wiley and Sons, Chichester, England, 2000.

K. Meister, M. Balmer, F. Ciari, A. Horni, M. Rieser, R. A. Waraich, and K. W. Axhausen. Large-scale agent-based travel demand optimization applied to switzerland, including mode choice. paper presented at the 12th World Conference on Transportation Research, July 2010.

D. K. Merchant and G. L. Nemhauser. A model and an algorithm for the dynamic traffic assignment problems. *Transportation Science*, **12**, 183–199, 1978*a*.

D. K. Merchant and G. L. Nemhauser. Optimality conditions for a dynamic traffic assignment model. *Transportation Science*, **12**(3), 200–207, 1978*b*.

E. Miller. Microsimulation and activity-based forecasting. *in* T. T. Institute, ed., 'Activity-Based Travel Forecasting Conference: Recommendations, and Compendium of Papers', pp. 151—-172. Travel Model Improvement Program, US Department of Transportation, US Environmental Protection Agency, June 1996.

F. Mosteller. Association and estimation in contingency tables. *Journal of the American Statistical Association*, **63**, 1–28, 1968.

K. Müller and K. W. Axhausen. Population synthesis for microsimulation: State of the art. *Presented at the 90th Annual Meeting of the Transportation Research Board*, 2011.

K. Nagel and G. Flötteröd. Agent-based traffic assignment: going from trips to behavioral travelers. *in* '12th International Conference on Travel Behaviour Research (IATBR)', Jaipur, 2009.

K. Nagel, R. L. Beckman, and C. L. Barrett. Transims for transportation planning. *in* 'In 6th Int. Conf. on Computers in Urban Planning and Urban Management'. Addison-Wesley, Reading,Massachusetts, 1999.

N. Oppenheim. *Urban Travel: From Individual Choices to General Equilibrium.* J. Wiley and Sons, New York, 1995.

J. D. Ortúzar and L. G. Willumsen. *Modelling Transport.* J. Wiley and Sons, Chichester (England), 3rd edn, 2001.

J. Pan, M. A. Khan, I. S. Popa, K. Zeitouni, and C. Borcea. Proactive vehicle re-routing strategies for congestion avoidance. *in* 'Distributed Computing in Sensor Systems (DCOSS), 2012 IEEE 8th International Conference on', pp. 265–272. IEEE, 2012.

S. Peeta and A. K. Ziliaskopoulos. Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and Spatial Economics*, **1**(3-4), 233–265, 2001.

A. J. Pel, M. C. J. Bliemer, and S. P. Hoogendoorn. A review on travel behaviour modelling in dynamic traffic simulation models for evacuations. *Transportation*, **39**(1), 97–123, 2012.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edn, 2007.

D. R. Pritchard and E. J. Miller. Advances in agent population synthesis and application in an integrated land use and transportation model. *Presented at the 88th Transportation Research Board Annual Meeting*, 2009.

M. Pursula. Simulation of traffic systems-an overview. *Journal of Geographic Information and Decision Analysis*, **3**(1), 1–8, 1999.

A. K. Rathi and Z. A. Nemeth. Freesim: A microscopic simulation model of freeway lane closures (abridgment). *Transportation Research Record*, **1091**, 1986.

P. Salvini and E. J. Miller. Ilute: An operational prototype of a comprehensive microsimulation model of urban systems. *Networks and Spatial Economics*, **5**, 217–234, 2005.

M. Scheutz, P. Schermerhorn, R. Connaughton, and A. Dingler. Swages–an extendable parallel grid experimentation system for large-scale agent-based alife simulations. *Proc. Artificial Life X*, 2006.

B.D. Spear. *Application of new travel demand forecasting techniques to transportation planning: a study of individual choice models*. Dept. of Transportation, Federal Highway Administration, Office of Highway Planning, Urban Planning Division, 1977.

S. Srinivasan, L. Ma, and K. Yathindra. Procedure for forecasting households characteristics for input to travel-demand models. Final Report TRC-FDOT-64011-2009, Transportation Research Center, University of Florida, FL, USA, 2008.

M. H. Ueberschaer. Choice of routes on urban networks for the journey to work. *Highway Research Record*, **369**, 1971.

K. H. van Dam, I. Nikolic, and Z. Lukszo. *Agent-based Modelling of Socio-technical Systems*, Vol. 9 of *Agent-Based Social Systems Series*. Springer-Verlag New York Incorporated, 2012.

D. Voas and P. Williamson. An evaluating goodness-of-fit measures for synthetic microdata. *Geographical & Environmental Modeling*, **5**(2), 177–200, 2001.

M. Wachs. Relationships between drivers'attitudes toward alternate routes and driver and route characteristics. *Highway Research Record*, **197**, 70–87, 1967.

P. Waddell. Urbansim: Modeling Urban Development for Land Use, Transportation and Environmental Planning. *Journal of the American Planning Association*, **3**(3), 297–314, 2002.

J. Wardrop. Some theoretical aspects of road traffic research. *Proceedings of the Institute of Civil Engineers, part II*, **1**, 325–378, 1952.

A. G. Wilson. *Entropy in urban and regional modelling*. Pion, London, 1970.

A. G. Wilson. *Urban and regional models in geography and planning*. J. Wiley and Sons, Chichester, England, 1974.

A. G. Wilson and C.E. Pownall. A new reprensentation of the urban system for modeling and for the study of micro-level interdependence. *Area*, **8**, 246–254, 1976.

M. Wooldridge. *An introduction to multiagent systems*. Wiley, 2008.

X. Ye, K. Konduri, R. M. Pendyala, B. Sana, and P. Waddell. A methodology to match distributions of both household and person attributes in the generation of synthetic populations. *in* 'TRB 88th Annual Meeting Compendium of Papers DVD', Washington, U.S.A., 2009. Transportation Research Board - 88th Annual Meeting.

D. Zielstra and H. H. Hochmair. Using free and proprietary data to compare shortest-path lengths for effective pedestrian routing in street networks. *Transportation Research Record: Journal of the Transportation Research Board*, **2299**(1), 41–47, 2012.